

Hoe requirements traceability het beste ingericht kan worden.

Onderzoek naar inrichting van Requirements traceability in theorie, de Military Standard 498 en de praktijk.



Studentnaam: Ruud Winkel
Studentnummer: 850477339
Datum: 18 juni 2013

The best way to organize requirements traceability

Research to organization of requirements traceability in theorie, the Military Standard 498 and in practice

Opleiding: Open Universiteit Nederland,
Faculteit Informatica Masteropleiding Business Process
Management and IT (T89317)

Afstudeercommissie:

1e begeleider: Dr. E.E. Roubtsova
2e begeleider: Dr. Anda Counotte
Examinator: Prof.Dr. Ir. Stef Joosten

Voorwoord

Voor u ligt het onderzoeksrapport dat ik ter afsluiting van mijn studie "Business Proces Management en IT" binnen de faculteit Informatica aan de Open Universiteit heb geschreven.

Het afstuderen in combinatie met een fulltime baan en een gezin met twee kinderen heeft veel doorzettingsvermogen gevraagd en had vaak een conflicterend effect met mijn rol als vader en echtgenoot. De periode van het afstuderen heeft dan ook zijn hoogte- en dieptepunten gekend. Het is een leerzaam maar vooral ook uitdagend traject geweest om te komen tot een waardevol onderzoek dat bijdraagt aan, zowel het wetenschappelijk vakgebied van informatie technologie (IT), alsook in praktische zin. De kennis die ik heb opgedaan van requirements traceability in combinatie met de Military Standard 498 zullen mij zeker ook in de rest van mijn carrière van nut zijn.

Bij dezen wil ik iedereen bedanken die een bijdrage geleverd heeft aan de totstandkoming van dit onderzoek. In het bijzonder wil ik mijn vrouw Andrea bedanken voor al haar geduld en energie die nodig waren voor mij om dit onderzoek tot een succesvol einde te brengen naast ons drukke gezinsleven.

Ook dank ik Rob van der Boog en Johan van der Linde voor de kans die zij mij hebben gegeven binnen ProRail om het empirisch onderzoek uit te voeren. Uiteraard dank ik ook mijn collega's die, waar gevraagd, mij van de nodige informatie en ideeën hebben voorzien.

Tot slot wil ik dr. Ella Roubtsova bedanken voor haar inzichten en feedback om bij te sturen waar mogelijk. Ook dank aan dr. Anda Counotte voor haar rol binnen dit afstudeertraject. Mede dankzij hen is het mogelijk geweest mijn afstudeeronderzoek goed af te sluiten.

Ik hoop dat u veel plezier beleeft aan het lezen van dit rapport.

Ruud Winkel

Woerden, 2013

Samenvatting

In de praktijk blijkt requirements management vandaag de dag nog steeds een uitdaging. Projecten leveren bijvoorbeeld niet op wat gevraagd is en waarom bepaalde zaken ontwikkeld worden is onduidelijk. Daarnaast is er een onbeheersbare stroom aan wijzigingen en leveren projecten te laat op. Helaas wordt de nut van het investeren in requirements traceability niet duidelijk in het begin van projecten maar meer in verdere fasen.

Dit empirisch onderzoek richt zich op het ICT-services domein van ProRail en dan specifiek op het gebied van ontwikkeling en beheer van applicatieve oplossingen. ICT-services is het domein van waaruit alle denkbare ICT dienstverlening, die een organisatie als ProRail ondersteunt, geleverd wordt. Bij verkenning van het domein is gebleken dat de projecten die er spelen nogal variëren in aanpak en dat de wijze waarop requirements management en in het bijzonder requirements traceability wordt toegepast uiteenloopt. Het blijkt dat er soms wel een standaard wordt toegepast voor documentatie en requirements management en soms niet. De standaard die soms toegepast wordt is de Military Standard 498. Regelmatig is er discussie over het product dat door het project opgeleverd wordt. Daarbij kan het gaan om requirements waarin in zijn geheel niet is voorzien of om wijzigingen waar niet alle belanghebbenden (stakeholders) zich in kunnen vinden. Dit leidt tot situaties waarbij extra kosten worden gemaakt om alsnog aan requirements te voldoen. Of tot projecten die uitlopen en dus extra kosten met zich mee brengen. Maar ook producten die uiteindelijk maar ten dele worden benut. In sommige gevallen leidt dit weer tot een nieuw project om alsnog het gewenste resultaat te realiseren.

Dit onderzoek zal zich richten op de wijze waarop requirements traceability wordt gerealiseerd. Zowel in de theorie, de documentatie standaard Military Standard 498 en in de praktijk. Het doel is om inzicht te krijgen in de wijze waarop requirements traceability het best kan worden ingericht.

Dit inzicht wordt verkregen door antwoord te geven op de volgende onderzoeksvraag:

Wat zijn de verschillen tussen de conceptuele modellen van requirements traceability in theorie, in de Military Standard 498 en in de praktijk en hoe kan traceability het best ingericht worden?

Er is een kwalitatief onderzoek uitgevoerd. Het onderzoek verkent de literatuur op gebied van requirements traceability, de Military Standard 498 en geeft in een conceptueel model weer op welke wijze requirements traceability in de Military Standard 498 wordt toegepast. Vervolgens wordt de requirements traceability getoetst aan een vijftal ICT projecten in de praktijk. Met behulp van conceptuele modellen zal inzichtelijk gemaakt worden wat de verschillen zijn tussen de theorie, Military Standard 498 en de praktijk. Met een conceptueel model bedoelen we een grafische weergave waarin alle relevante elementen (concepten) en hun relaties worden weergegeven. De verwachting is dat de resultaten kunnen bijdragen aan inzicht in de wijze waarop requirements traceability het best gerealiseerd kan worden. Het onderzoek bevat dan ook conclusies en aanbevelingen op dit gebied.

Aan requirements management worden in de theorie en de praktijk nogal eens verschillende betekenissen gegeven. Het lijkt daarom wenselijk om helderheid te verkrijgen over wat we onder requirements management verstaan. Op basis van beschrijvingen in de literatuur hebben we kunnen vaststellen dat onderzoekers er over de inhoud van requirements management verschillende opvattingen op na houden. Het verschil zit hem hoofdzakelijk in welke processtappen of activiteiten men onder requirements management schaaft. Er zijn onderzoekers die de definitie van requirements management beschrijven als: 'requirements management is het verzamelen, identificeren, onderhouden, analyseren, traceren en het managen van de wijzigingen op de requirements'. Anderen scharen activiteiten als het verzamelen, identificeren onder requirements development en noemen als requirements management activiteiten alleen de traceerbaarheid en het managen van wijzigingen.

Dit onderzoek hanteert de definitie: 'The goal of requirements management is to capture, store, disseminate, and manage information. Requirements management includes all activities concerned with change & version control, requirements tracing, and requirements status tracking'. Deze definitie geeft bondig een opsomming van de activiteiten, die in vrijwel elke definitie in ieder geval genoemd

worden. Voor deze definitie is gekozen omdat dit afstudeeronderzoek het begrip requirements management slechts verkent en zich richt op requirements traceability bij projecten in de theorie, praktijk en requirements traceability in een software ontwikkel- en documentatie standaard. Het onderzoek heeft niet als doel om tot een nieuwe definitie van requirements management te komen dan wel dit gebied te onderzoeken.

Requirements hebben hun oorsprong maar maken gedurende de ontwikkeling van een systeem ook een ontwikkeling door. Requirements traceability maakt het mogelijk om de oorsprong en ontwikkeling te herleiden en te beschrijven. Dit is belangrijk om requirements te begrijpen en om te verifiëren dat aan alle requirements is voldaan. Ook kan requirements traceability informatie geven over de rechtvaardiging, belangrijke besluiten en aannames die ten grondslag liggen aan de requirements. Uit diverse onderzoeken in de literatuur blijkt het nut en de noodzaak van requirements traceability. Requirements traceability is een belangrijk mechanisme om kwaliteit te waarborgen en het biedt ondersteuning bij de continue wijzigingen waaraan requirements onderhevig zijn. Ook is het belang van requirements traceability erkend in standaarden als ISO/IEC 12207 en IEEE Standard 830-1998. Met behulp van requirements traceability kunnen de problemen zoals producten die niet voldoen of wijzigingen waar niet elke belanghebbende baat bij heeft voorkomen worden.

Dit onderzoek hanteert als definitie van requirements traceability *“refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases).”*. Voor deze definitie is gekozen omdat dit afstudeeronderzoek zich richt op requirements traceability bij projecten in de theorie, praktijk en requirements traceability in een software ontwikkel en documentatiestandaard. Het onderzoek heeft niet als doel om tot een nieuwe definitie van traceability te komen dan wel dit gebied te onderzoeken.

Op basis van het literatuur onderzoek naar requirements traceability is een conceptueel model gekozen wat als uitgangspunt gehanteerd wordt voor de minimale inrichting van requirements traceability.

De Military Standard 498 is een software ontwikkel- en documentatie standaard. De standaard is ontstaan uit het samenvoegen van de standaarden Department of Defense Standard 2167A en Department of Defense Standard 7935A met een aantal verbeteringen. Één van de voornaamste hiervan is de verbeterde comptabiliteit met incrementele en evolutionaire standaarden. De standaard beschrijft het ontwikkel proces en alle bijbehorende documentatie. Het omvat 22 Data Item Descriptions die beschrijven wat ieder document in dit proces moet bevatten.

De Military Standard 489 besteedt aandacht aan het borgen van requirements traceability. De standaard legt nadrukkelijk de verantwoordelijkheid voor requirements traceability mede bij de ontwikkelaar. Niet alleen met betrekking tot specificatie en ontwerp maar ook voor testcases, CSCI requirements, system requirements, CSCI source files and software units. De standaard vereist ten behoeven van requirements traceability; identificatie voor requirements en dat alle relevante items opgenomen worden in de daartoe bestemde secties van de documentatie. De standaard benoemt dit in het algemeen, maar ook expliciet voor de onderdelen:

- System/Subsystem specification (SSS)
- System/Subsystem Design Description (SSDD)
- Software Requirements Specification (SRS)
- Interface Requirements Specification (IRS)
- Software Design Description (SDD)
- Interface Design Description (IDD)
- Database Design Description (DBDD)

Op basis van het bestuderen van de Military Standard 498 en de elementen die het biedt om requirements traceability in te richten is een conceptueel model opgesteld. Dit model is een weergave van de objecten die een rol spelen bij het inrichten van requirements traceability en de relaties hiertussen.

Uit de praktijk zijn de documenten van vijf projecten geanalyseerd op de requirements traceability. Dit zijn zowel projecten die met een standaard werken als projecten die geen standaard hebben

gehanteerd op het gebied van requirements management. Per project is een tweetal requirements getraceerd om zo zichtbaar te maken in welke mate requirements traceability in een project gerealiseerd is. Op basis van de resultaten uit het literatuuronderzoek en empirisch onderzoek is een vergelijking gemaakt ten aanzien van de mate van requirements traceability. Daarbij is ook gekeken naar het verschil tussen de realisatie van requirements traceability in de literatuur en in de Military Standard 498. Het blijkt dat deze allebei sterk overeenkomen in mate van requirements traceability en de wijze waarop dit gerealiseerd wordt.

Uit de analyse van projecten blijkt dat er een behoorlijk verschil is tussen de mate van requirements traceability die gerealiseerd is in projecten die de Military Standard 498 volledig hebben toegepast, projecten die de standaard alleen 'in naam' hebben toegepast en projecten die geen standaard hebben toegepast. Met in 'naam' toegepast bedoelen we niet volledig uitgevoerd volgens de richtlijnen van de standaard.

Met behulp van het inzicht in de wijze waarop de literatuur requirements traceability realiseert en hoe de Military Standard 498 dit doet kunnen we komen tot de volgende conclusie.

Daar waar de standaard wordt toegepast en de handvatten voor traceability worden ingevuld leidt dit zeker tot verbetering van requirements traceability in het project. Om requirements traceability te realiseren helpen de handvatten die de standaard biedt; zoals een unieke identificatie van requirements en het opstellen van een traceability hoofdstuk of matrix. Dit onderzoek toont aan dat voor het goed inrichten van requirements traceability, het onderkennen van elementen als requirements, systemen- en subsystemen, externe systemen, test- en verificatieprocedures essentieel zijn. Daarbij is het vooral van belang dat ook de relaties tussen deze elementen onderkend worden en het feit dat requirements in al deze elementen terugkeren. Pas als dit erkend is en ook bij het managen van de requirements hiermee rekening gehouden wordt, kan traceability gerealiseerd worden.

Dit onderzoek levert hiermee in zijn algemeenheid een bijdrage aan het inzicht in de wijze waarop requirements traceability tot stand komt en de verschillen hierin tussen theorie en praktijk. Voor de praktijk levert het inzicht op welke aspecten belangrijk zijn om de requirements traceability te verbeteren.

Inhoud

Voorwoord	2	
Samenvatting	3	
1 Inleiding	10	
1.1 Aanleiding		10
1.2 Doelstelling		10
1.3 Probleemstelling		10
1.4 Onderzoeksvraag		11
1.5 Deelvragen literatuurstudie		11
1.6 Onderzoeksstrategie		11
1.7 Deelvragen empirisch onderzoek		12
1.8 Onderzoeksmodel		12
1.9 Leeswijzer		13
2 Theoretisch kader	14	
2.1 Requirements management		14
2.1.1 Definitie		14
2.1.2 Conclusie deelvraag 1		15
2.2 Requirements Traceability		15
2.2.1 Definitie requirements traceability		15
2.2.2 Referentie model requirements traceability		16
2.2.3 Conclusie deelvraag 2		18
2.3 Software ontwikkeling en documentatie standaarden		19
3 Military Standard 49820		
3.1 Requirements traceability Military Standard 498		24
3.2 Data Item Descriptions in relatie tot de standaard		24
3.3 Conclusie deelvraag 3		26
4 Analyse van requirements traceability in projecten	29	
4.1 Project en domein ASTRIS		30
4.1.1 Requirements structuur ASTRIS		30
4.1.2 Requirements traceability project ASTRIS		30
4.2 Resultaten analyse project ASTRIS		37
4.3 Project en domein BBMS		39
4.3.1 Requirements structuur BBMS		39
4.3.2 Requirements traceability project BBMS		40
4.4 Resultaten analyse project BBMS		42
4.5 Project en domein SBG++		42
4.5.1 Requirements structuur SBG++		44
4.5.2 Requirements traceability SBG++		44
4.6 Resultaten analyse project SBG++		46
4.7 Project en domein Routelint		47

4.7.1	Requirements structuur Routelint	48
4.7.2	Requirements traceability Routelint	48
	System/Subsystem Specification	51
	System/Subsystem Design Description	51
4.8	Resultaten analyse project Routelint	53
4.9	Project en domein RRCB	55
4.9.1	Requirements structuur RRCB	56
4.9.2	Requirements traceability RRCB	56
4.10	Resultaten analyse project RRCB	60
4.11	Resultaten analyse	61
5	Conclusie en aanbevelingen	66
5.1	Vervolg onderzoek.	67
5.2	Aanbevelingen	67
	Bibliografie	68
	Bijlage I ASTRIS Requirements	70
	System/Subsystem Specification – ASTRIS	70
	System/Subsystem Design Description	72
	SRS beveiligingscomponent	73
	Bijlage II BBMS Requirements	76
	BBMS Functioneel Ontwerp	91
	Bijlage III Requirements SBG++	94
	Operational Concept Description	94
	Software Requirements Specification	94
	Interface Requirements Specification	96
	Bijlage IV Requirements Routelint	98
	Operational Concept Description	98
	System/Subsystem Specification	98
	Software Requirements Specification	99
	System/Subsystem Design Description	100
	Bijlage V Requirements RRCB	101
	Use Cases RRCB	102
	System/Subsystem Specification	103
	Interface Requirements Specification	104

Figuur 1 "Onderzoeksmodel afstudeeronderzoek"	13
Figuur 2 Traceability meta model. Bron: Ramesh, B en M. Jarke (2001)	17
Figuur 3 Low End traceability model. Bron: Ramesh, B en Jarke (2001)	18
Figuur 4 Military Standard 498 Data Item Descriptions. Bron: Norma A. Stopyra, (1996).	20
Figuur 5 Key features of three DOD program strategies. Bron: Military Standard 498, http://www.everyspec.com.....	21
Figuur 6 Weergave ontwikkel proces Military Standard 498. Bron: Abelia.com	23
Figuur 7 'Military Standard 498 Software Development Activities'. Bron: Military Standard 498, http://www.everyspec.com.....	25
Figuur 8 'Military Standard 498 Data requirements'. Bron: Military Standard 498, http://www.everyspec.com.....	26
Figuur 9 'Conceptueel model Requirements Traceability Military Standard 498'	27
Figuur 10 'Overzicht van projecten, toepassing Military Standard 498 en getraceerde requirements'. ..	29
Figuur 11 'Schematische weergave traceability relaties ASTRIS systeemeisen'. Bron: ASTRIS- traceabilitymatrix.....	31
Figuur 12 'Relaties ASTRIS systeemeisen'. Bron: ASTRIS-traceabilitymatrix.....	31
Figuur 13 'Inleiding en uitleg traceability matrix' Bron: ASTRIS-traceabilitymatrix.....	32
Figuur 14 'Traceability systeem eisen'. Bron: ASTRIS-traceabilitymatrix.	33
Figuur 15 Systeemeis ASTRIS-UC-VHR uit System/Subsystem Specification'. Bron: ASTRIS System/Subsystem Specification.	33
Figuur 16 Testcase 'ASTRIS-TG-VHR'. Bron: ASTRIS-STD-KT.....	34
Figuur 17 'Software eis EMC-IF-09'. Bron: ASTRIS-ELEMENTCOMP-SRS.....	34
Figuur 18 'Software eis EMC-FUNC-004'. Bron: ASTRIS-ELEMENTCOMP-SRS.....	35
Figuur 19 'Software eis EC-DATA-07'. Bron: ASTRIS-ELEMENTCOMP-SRS.	35
Figuur 20 'Traceability component eisen naar testdocumentatie'. Bron: ASTRIS-Traceabilitymatrix... ..	35
Figuur 21 'Testgeval EMC-TG-PARC'. Bron: ASTRIS-ELEMENTCOMP-STD.	35
Figuur 22 'Component requirements relatie met ontwerp'. Bron: Traceabilitymatrix project ASTRIS. .	36
Figuur 23 'Component element service in SDD'. Bron: SDD project ASTRIS.	36
Figuur 24 'Systeem eis ASTRIS-ROBU-6 & gerelateerde Interface eis I-BEH-ALGEMEEN-ROBUUST- 2'. Bron: ASTRIS-traceabilitymatrix & ASTRIS_NXA-BEHEER-IRS.....	37
Figuur 25 'Traceability model project ASTRIS.	38
Figuur 26 'Requirements structuur BBMS'. Bron:.....	39
Figuur 27 'Pakket van eisen BBMS'. Bron: PvE 1.0 BBMS.....	40
Figuur 28 'Functioneel ontwerp BBMS'. Bron: FO dekkingslagen BBMS.	41
Figuur 29 'BBMS requirement GIS Viewer'. Bron: PvE 1.0 BBMS.....	41
Figuur 30 'BBMS functioneel ontwerp GIS Viewer'. Bron: Functionele Specificatie gisviewer 1.9	42
Figuur 31 'Traceability model project BBMS'.....	42
Figuur 32 'SBG+++ relaties buitenwereld'. Bron: SRS SBG++ 2.0 g57.....	43
Figuur 33 'Voorbeeld Software Requirements Specificatie wijze van User Stories Identificatie' Bron: SRS SBG++.....	45
Figuur 34 'Voorbeeld identificatie User Stories in SRS en requirements in IRS'. Bron: SRS & IRS SBG++.....	46
Figuur 35 'Traceability model project SBG++'.....	47
Figuur 36 'Traceability System Requirements & Software Requirements Routelint. Bron: OOGST- SSS-040.pdf & OOGST-DST-SRS-030-renvooi.pdf Routelint.	49
Figuur 37 'Requirement RMR.CD.2 gerelateerd aan Software CI'. Bron: Routelint SSDD.....	49
Figuur 38 'Requirement RMR.CD.2 gerelateerd aan Interface Requirements'. Bron: IRS project Routelint.....	50
Figuur 39 'Requirements Specificatie Routelint'. Bron: SRS project routelint.....	50
Figuur 40 'Traceability tabel SRS Routelint' Bron: SRS project routelint.	51
Figuur 41 'Aan Software Requirements Specificatie gerelateerd eis'. Bron: SSS project Routelint. ...	51
Figuur 42 'Ontwerp SSDD gerelateerd aan requirement EI.GEBR.1'. Bron: SSDD project Routelint.	51
Figuur 43 'Traceability in SSDD gerelateerd aan eis EI.GEBR.1'. Bron: SSDD project Routelint.	52
Figuur 44 'Eis EI.GEBR.1' gerelateerd aan testgevallen. Bron: OOGST-P210-DST-OFE-STD-IT	52
Figuur 45 'Testgeval gerelateerd aan EI.GEBR.1'. Bron: OOGST-P210-DST-OFE-STD-IT	53
Figuur 46 'Traceability model project Routelint'.....	54
Figuur 47 'Regelkring PGO-contract'. Bron: Project Start Architectuur RRCB.	55
Figuur 48 'High level requirement RRCB'. Bron: High Level Requirements RRCB versie 13 okt.	56
Figuur 49 'Use Case model RRCB' Bron: UseCasemodel en Business Requirements RRCB.	57
Figuur 50 'Requirement in SSS' Bron: RRCB SSS 1.0	58

Figuur 51 'High Level Requirement 14 RRCB'. Bron: High Level Requirements RRCB.	58
Figuur 52 'Use Case 9 RRCB'. Bron: Use Case model en Business Requirements RRCB.	58
Figuur 53 'Interface eisen RRCB'. Bron: IRS RRCB.	59
Figuur 54 'Stakeholders Use Cases project RRCB'. Bron: Key users RRCB per usecase.docx.....	60
Figuur 55 'Conceptueel model requirements traceability RRCB'.	61
Figuur 56 'Vergelijking traceability objecten theorie, Military Standard 498 en projecten'	62
Figuur 57 'Vergelijking traceability relaties theorie, Military Standard 498 en projecten'.	63
Figuur 58 'Productboom ASTRIS'. Bron: SSS project ASTRIS.	70
Figuur 59 'Systeem context'. Bron: ASTRIS-SSS.....	70
Figuur 60 'Use case thema'. Bron: ASTRIS-SSS.....	71
Figuur 61 'Domeinmodel'. Bron: ASTRIS-SSDD.....	73
Figuur 62 'Service operatie 'VerhinderRijwegElement'. Bron: ASTRIS-SRS.....	74
Figuur 63 'Definitie verhinderRijwegElement'. Bron: ASTRIS-SRS.	74
Figuur 64 'Uitwerking verhinderRijwegElement service request'. Bron: STD project ASTRIS.	75
Figuur 65 "Voorbeeld Data Item Description Operational Concept Description"	98
Figuur 66 "Operational Concept Description Routelint"	98
Figuur 67 'SSS Routelint voorbeeld Prefix'	99
Figuur 68 'SRS Routelint'	99
Figuur 69 "Ontwerpkeuze met referentie naar SSS"	100
Figuur 70 "Architectuur ontwerp met verwijzing naar SSS.....	100
Figuur 71 'Use Case Diagram RRCB'	102
Figuur 72 'Voorbeeld Use Case project RRCB'.....	103
Figuur 73 'System/Subsystem Specification RRCB'	104
Figuur 74 'Interface RRCB'	105

1 Inleiding

In dit hoofdstuk wordt aandacht besteed aan de aanleiding voor dit onderzoek. Vervolgens komen fundamentele onderdelen als de doelstelling, probleemstelling en onderzoeksvragen aan bod. Daaropvolgend zal inzicht worden verschaft in de onderzoeksstrategie en het onderzoeksmodel. Het hoofdstuk wordt afgesloten met een leeswijzer.

1.1 Aanleiding

Dit empirisch onderzoek richt zich op het ICT-services domein van ProRail en dan specifiek op het gebied van ontwikkeling en beheer van applicatieve oplossingen. ICT-services is het domein van waaruit alle denkbare ICT dienstverlening die een organisatie als ProRail ondersteunt, geleverd wordt. Bij verkenning van het domein is gebleken dat de projecten die er spelen nogal variëren in aanpak en dat de wijze waarop requirements management en in het bijzonder requirements traceability wordt toegepast uiteenloopt. Het blijkt dat er soms wel een standaard wordt toegepast voor documentatie en requirements management en soms niet. De standaard die soms toegepast wordt is de Military Standard 498.

Gezien er zoveel verschil zit in de wijze waarop omgegaan wordt het vastleggen van requirements bestaat het vermoeden dat de requirements veelal niet traceerbaar zijn. De verwachting is echter wel dat requirements traceability beter is ingericht bij de projecten waarbij de Military Standard 498 wel is toegepast. Mede omdat uit de verkenning is gebleken dat er bij deze projecten meer requirements documentatie beschikbaar was. Het lijkt interessant om te onderzoeken wat de literatuur beschrijft over het inrichten van requirements traceability. Hoe de Military Standard 498 om gaat met requirements traceability en hoe projecten met en zonder Military Standard 498 uitgevoerd, hier in de praktijk mee omgaan.

1.2 Doelstelling

Dit onderzoek richt zich op requirements management en specifiek op requirements traceability. Er is in de literatuur al veel te lezen over deze onderwerpen. Het zijn complexe onderwerpen die van groot belang zijn bij het ontwikkelen van software. Dit onderzoek richt zich op de requirements traceability in de literatuur, in de Military Standard 498 en in de praktijk. Interessant is om te bekijken wat nu de verschillen zijn tussen de wijze waarop de literatuur beschrijft hoe requirements traceability gerealiseerd kan worden, hoe de Military Standard 498 requirements traceability bereikt en hoe projecten in de praktijk daarmee omgaan.

De verwachting is dat de resultaten kunnen bijdragen aan inzicht in de wijze waarop requirements traceability het beste gerealiseerd kan worden.

1.3 Probleemstelling

In de dagelijkse praktijk van de ICT ontstaan vaak nog problemen omdat bijvoorbeeld niet goed is vastgelegd wat de requirements waren, hoe deze zijn ontstaan of aan welke wijzigingen de requirements onderhevig waren. Het resultaat kan bijvoorbeeld zijn dat een product niet aan de verwachtingen voldoet, maar ook een project dat uitloopt omdat de impact van een wijziging niet was voorzien.

In het onderzoeksdomein ICT Services van ProRail zijn de resultaten van projecten vaak wisselend. Er zit duidelijk veel verschil in aanpak per project op het gebied van requirements management en in het bijzonder requirements traceability. Regelmatig ontstaat er discussie over het product dat door het project opgeleverd wordt. Daarbij kan het gaan om requirements waarin in zijn geheel niet is voorzien of om wijzigingen waar niet alle belanghebbenden (stakeholders) zich in kunnen vinden. Dit leidt tot situaties waarin extra kosten worden gemaakt om alsnog aan requirements te voldoen. Tot projecten die uitlopen en dus extra kosten met zich mee brengen. Maar ook producten die uiteindelijk maar ten dele worden benut. In sommige gevallen leidt dit dan weer tot een nieuw project om alsnog het gewenste resultaat te realiseren.

Requirements hebben hun oorsprong, maar maken gedurende de ontwikkeling van een systeem ook een ontwikkeling door. Requirements traceability maakt het mogelijk om de oorsprong en ontwikkeling te herleiden en te beschrijven. Dit is belangrijk om requirements te begrijpen en om te verifiëren dat aan alle requirements is voldaan. Ook kan requirements traceability informatie geven over de rechtvaardiging, belangrijke besluiten en aannames die ten grondslag liggen aan de requirements.

(Ramesh, 2001). Hiermee kunnen de problemen zoals producten die niet voldoen of wijzigingen waar niet iedere belanghebbende baat bij heeft voorkomen worden.

1.4 Onderzoeksvraag

Dit onderzoek zal zich richten op de wijze waarop requirements traceability wordt gerealiseerd. Zowel in de theorie, de documentatie standaard Military Standard 498 als in de praktijk. Het doel is om inzicht te krijgen in de wijze waarop requirements traceability het beste ingericht kan worden.

De onderzoeksvraag luidt dan ook als volgt:

Wat zijn de verschillen tussen de conceptuele modellen van requirements traceability in theorie, in de Military Standard 498 en in de praktijk en hoe kan traceability het best ingericht worden?

1.5 Deelvragen literatuurstudie

Om antwoord te krijgen op deze onderzoeksvraag is het noodzakelijk dat we inzicht krijgen in het domein. Daarom is in de literatuurstudie de aandacht uitgegaan naar de volgende deelvragen.

- 1) Wat wordt verstaan onder requirements management?
- 2) Wat is requirements traceability?
- 3) Wat is de Military Standard 498 en hoe gaat de Military Standard 498 om met requirements traceability?

1.6 Onderzoeksstrategie

Er is een kwalitatief onderzoek uitgevoerd. Het onderzoek verkent de literatuur op gebied van requirements traceability, de Military Standard 498 en geeft in een conceptueel model weer op welke wijze requirements traceability in de Military Standard 498 wordt toegepast. Vervolgens wordt de requirements traceability getoetst aan ICT projecten in de praktijk. Met behulp van conceptuele modellen zal inzichtelijk gemaakt worden wat de verschillen zijn tussen de theorie, Military Standard 498 en de praktijk. Met een conceptueel model bedoelen we een grafische weergave waarin alle relevante elementen (concepten) en hun relaties worden weergegeven.

Op basis van deze conceptuele modellen wordt duidelijk hoe de requirements traceability wordt gerealiseerd. Hieruit kunnen een aantal conclusies worden getrokken die inzicht bieden hoe de Military Standard 498 met requirements traceability omgaat en hoe requirements traceability het beste gerealiseerd kan worden.

Uit diverse onderzoeken in de literatuur blijkt het nut en de noodzaak van requirements traceability. Requirements traceability is een belangrijk mechanisme om kwaliteit te waarborgen en biedt ondersteuning bij de continue wijzigingen waaraan requirements onderhevig zijn (Gotel, 95) (Gotel & Finkelstein, 1994). Ook is het belang van requirements traceability erkend in standaarden als ISO/IEC 12207 en IEEE Standard 830-1998. Daarom is eerst het gebied van requirements management en requirements traceability verkend. Vervolgens is de Military Standard 498 onderzocht en de wijze waarop de standaard omgaat met requirements traceability. De bevindingen zijn weergegeven in een conceptueel model.

In het empirisch onderzoek wordt de requirements documentatie van een vijftal projecten geanalyseerd waarbij per project een tweetal requirements is getraceerd. Het aantal projecten en getraceerde requirements is gekozen uit praktische overwegingen als tijd en middelen. Er moet binnen afzienbare tijd een groot aantal documenten verzameld en bestudeerd worden. Wel is in het kader van betrouwbaarheid een minimum gesteld van twee projecten waarbij de Military Standard 498 toegepast werd en een tweetal waarbij dat niet het geval was. Hierbij is onderzocht in welke mate en op welke wijze traceability aanwezig is. Dit is gedaan door de project documentatie nauwgezet te bestuderen en daadwerkelijk traceability op specifieke requirements te toetsen. Dit door een tweetal requirements door middel van de project documentatie te traceren en te controleren of een requirement in het ontwerp ook terug te relateren is aan een requirement in de specificaties. Maar bijvoorbeeld ook door

te kijken of vastgelegd is welke andere requirements aan deze specifieke requirement gerelateerd zijn en of deze ook daadwerkelijk te herleiden waren.

De twee requirements die per project getraceerd zijn, zijn steekproefsgewijs gekozen. Dit is bewust gedaan. Door betrokkenen te raadplegen bij de keuze van de requirement is er sprake van mogelijke beïnvloeding door extra kennis. Requirements traceability heeft nu juist ten doel te realiseren dat ook iedereen zonder verregaande projectkennis de requirement kan traceren. Wel is bij de keuze voor de requirements als uitgangspunt de requirements specificatie gekozen. Er wordt tenslotte eerst gespecificeerd en pas dan ontworpen of getest. Ook is rekening gehouden met het feit dat een requirement bijvoorbeeld wel in een systeemontwerp terug kan komen maar niet in een interface ontwerp.

Van de verschillende wijzes waarop een requirement traceerbaar was, is per project een conceptueel model van de traceerbaarheid opgesteld. Hierdoor wordt niet alleen inzichtelijk hoe en in welke mate de Military Standard 498 requirements traceability realiseert in ICT projecten. Maar ook hoe en in welke mate, projecten die deze standaard niet hebben gebruikt, requirements traceability realiseren.

Op basis van deze resultaten wordt een vergelijking gemaakt tussen de conceptuele modellen en daarmee de traceerbaarheid van de requirements.

1.7 Deelvragen empirisch onderzoek

Om inzicht te krijgen in de wijze waarop projecten in de praktijk requirements traceability realiseren worden een vijftal projecten onderzocht waarvan een tweetal requirements wordt getraceerd. Dit zijn zowel projecten waarbij de Military Standard 498 is toegepast als projecten waarbij geen standaard is toegepast. Bij het onderzoeken van deze projecten zijn de volgende vragen beantwoord.

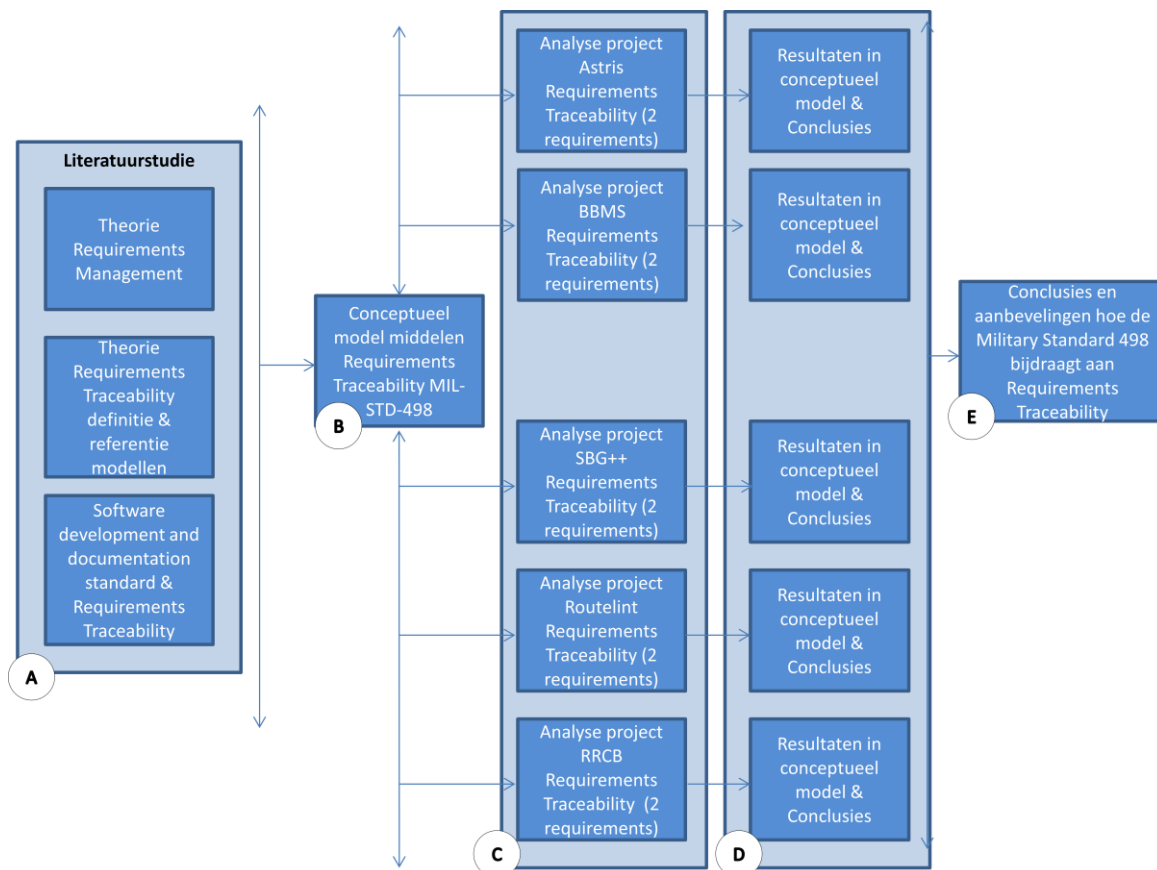
- 1) In welke mate is requirements traceability aanwezig bij de projecten in het domein van ICT services?
- 2) Hoe is requirements traceability ingericht in projecten waarbij geen standaard is toegepast?
- 3) Hoe is requirements traceability ingericht in projecten waarbij de Military Standard 498 is toegepast?

1.8 Onderzoeksmodel

Door het bestuderen van wetenschappelijke literatuur op het gebied van requirements management ontstaat inzicht in requirements traceability en de ontwikkel- en documentatiestandaard Military Standard 498. De conclusies die hieruit getrokken worden, kunnen op empirische wijze in de praktijk getoetst worden. De bevindingen en resultaten uit dit onderzoek maken de verschillen inzichtelijk tussen traceability in theorie, de Military Standard 498 en de praktijk. Op basis hiervan kunnen conclusies worden getrokken ten aanzien van de bijdrage van een standaard als de Military Standard 498 aan requirements traceability.

In het conceptueel model in Figuur 1 wordt het e.e.a. als volgt weergegeven:

- A. Literatuuronderzoek op het gebied van requirements management, requirements traceability, de Military Standard 498 en de wijze waarop de standaard requirements traceability inricht.
- B. Conceptueel model welke inzicht geeft in de middelen waarmee de Military Standard 498 requirements traceability realiseert.
- C. Empirisch onderzoek waarbij een vijftal projecten en per project een tweetal requirements worden getraceerd. Deze requirements zijn steekproefsgewijs geselecteerd.
- D. Conclusies op basis van het traceren van de requirements en het opstellen van een conceptueel model van de requirements traceability van het betreffende project.
- E. Conclusies en aanbevelingen op basis van de resultaten uit het onderzoek.



Figuur 1 "Onderzoeksmodel afstudeeronderzoek"

1.9 Leeswijzer

Dit verslag is opgebouwd uit vijf hoofdstukken. Het eerste hoofdstuk bevat de inleiding en besteedt aandacht aan de probleemstelling, onderzoeksvragen en de aanpak. Daarna zal het verslag de hoofdlijnen van het onderzoek volgen. Hoofdstuk 2 en 3 schetsen het theoretisch kader gevormd door de literatuurstudie en beantwoorden de deelvragen. Hoofdstuk 3 is daarbij specifiek gewijd aan de Military Standard 498 gezien het belang van deze standaard in het onderzoek. Hoofdstuk 4 geeft een beschrijving van het empirisch onderzoek waarbij een vijftal projecten uit de praktijk zijn geanalyseerd. Per project wordt in de eerste paragraaf een inleiding gegeven van het domein en type project. In de tweede paragraaf wordt op hoofdlijnen beschreven op welke wijze requirements in het project zijn vastgelegd. In de laatste paragraaf wordt het traceren van een tweetal requirements getoond. Hoofdstuk 5 gaat in op de resultaten van het onderzoek en de conclusies die hieruit getrokken kunnen worden. Het geeft een aanbeveling voor de praktijk en er worden suggesties gedaan voor vervolgonderzoek. In bijlage I worden voorbeelden van de verschillende requirements documentatie van project ASTRIS weergegeven, bijlage II toont een voorbeeld van het pakket van eisen ten aanzien van project BBMS, bijlage IV toont voorbeelden van de requirements documentatie van project Routelint en bijlage V toont de voorbeelden van de requirements documentatie van project RRCB.

Dit verslag is geschreven voor een breed publiek: zowel geïnteresseerden uit het wetenschappelijk vakgebied als voor professionals uit het vakgebied van de informatie technologie (IT).

Requirements traceability en de Military Standard 498 op zichzelf zijn niets nieuws. Echter het toepassen van requirements traceability is vandaag de dag nog steeds niet vanzelfsprekend in de IT Requirements management. Het realiseren van traceability is een complex en lastig proces. Software ontwikkeling resulteert in een niet-tastbaar product, waardoor het resultaat en de kwaliteit ervan niet zo snel getoetst kan worden. Het eindproduct voldoet vaak niet aan wat de aanvrager/klant aanvankelijk voor ogen had. Requirements traceability kan er toe bijdragen om dit te voorkomen. Het is dan ook interessant om te zien hoe in het IT vakgebied requirements traceability wordt toegepast en hoe dit zich verhoudt ten opzichte van de theorie en de Military Standard 498. Dit onderzoek draagt bij aan dit inzicht en het begrip van requirements traceability in ICT projecten.

2 Theoretisch kader

Dit hoofdstuk beschrijft de bevindingen van de literatuurstudie. Deze literatuurstudie is uitgevoerd om begrip te krijgen in wat requirements traceability betekent, met zich mee brengt en op welke wijze de software ontwikkel en documentatie standaard 'Military Standard 498'

(<http://www.abelia.com/498pdf/roadmap.pdf>) omgaat met requirements traceability.

2.1 Requirements management

Zowel in de theorie als in de praktijk worden er nogal eens verschillende betekenissen gegeven aan requirements management. Het lijkt daarom wenselijk om helderheid te verkrijgen over wat we onder requirements management verstaan. Hiervoor zijn verschillende definities uit de literatuur bekeken.

2.1.1 Definitie

Peatsch, Eberlein en Maurer (2003) geven aan: 'The goal of requirements management is to capture, store, disseminate, and manage information. Requirements management includes all activities concerned with change & version control, requirements tracing, and requirements status tracking.' Wat uitgebreider is de definitie van Davis (2004) 'Requirements management is de set van de activiteiten van het verzamelen van requirements, beslissen welke de juiste zijn om te realiseren en dit documenteren. Requirements elicitation is een subset van requirements management waarmee kandidaat requirements worden verzameld van klanten, gebruikers, onderwerp specifieke experts en andere stakeholders. Requirements triage is een subset waarbij vastgesteld wordt aan welke requirements voldaan moet worden geanalyseerd in de context van beschikbare ontwikkel resources, time to market, tussentijdse doelen en return on investment. Requirements specification is de subset waarbij het externe gedrag van het gewenste systeem wordt gedocumenteerd'.

Parviainen en Tihinen (2003) beschouwen requirements management naast requirements development als hoofdactiviteit van requirements engineering. Hierbij beschrijven ze requirements management als hoofdactiviteit die de subactiviteiten 'maintenance, identification, traceability and changemanagement' omvat.

Ook beschrijven ze requirements management als een 'continues and cross-section' activiteit binnen het requirements engineering. Het verzamelen van requirements scharen zij onder requirements development. Ook Pandey en Suman (2010) benoemen requirements management als zodanig. Zij benoemen echter alleen traceability en change management als activiteiten van requirements management. Wel geven zij nog aan dat requirements management soms het meest complexe requirements engineering proces is.

Als we kijken naar het onderzoek naar het specificeren van requirements door modellen gedreven van Santos Soares en Vrancken (2008), dan zien we dat ook zij requirements management naast requirements development als onderdeel van requirements engineering benoemen. Hierbij beschrijven zij requirements management als een beperkter onderdeel dat met name onderhoud zoals tracing en change management betreft. Elicitation bijvoorbeeld zien zij als requirements development.

2.1.1.1 CMMI

In het kader van een meer volwassen requirements management proces kan het Capability Maturity Model Integration (CMMI) ook een bepaalde kijk op requirements management bieden. CMMI is een benadering voor proces verbetering in organisaties. CMMI is het resultaat van een samenwerking tussen onderdelen uit de industrie, overheid en de Software Engineering Institute (SEI). Software ontwikkeling en requirements management als 'Key Process Area' van het CMMI model omschrijft requirements management als volgt: 'The activity of requirements management is focused on the control of the requirements gathering, establishing an agreement between the customer and the software team on the requirements, checking, reviewing and managing the changes on requirements. This activity is the process of ensuring that a software product produced from a set of requirements, will meet those requirements.' (Loconsole, 2001). De enige manier om een goede controle op requirements te krijgen is controle te houden op de continue veranderende definitie van een requirement gedurende de life cycle van het software product. Dit kan door de status van requirements vast te leggen en de stabiliteit te onderzoeken door middel van verschillende vragen. Hierdoor wordt namelijk het aantal wijzigingen op requirements en de kosten hiervan inzichtelijk.

2.1.2 Conclusie deelvraag 1

Op basis van deze beschrijvingen in de literatuur kunnen we vaststellen dat de onderzoekers er verschillende opvattingen op na houden over de inhoud van requirements management. Het verschil zit hem hoofdzakelijk in welke processtappen of activiteiten men onder requirements management schaaft. Er zijn onderzoekers die de definitie van requirements management beschrijven als: 'requirements management is het verzamelen, identificeren, onderhouden, analyseren, traceren en het managen van de wijzigingen op de requirements'. Anderen scharen activiteiten als het verzamelen, identificeren onder requirements development en noemen als requirements management activiteiten alleen de traceerbaarheid en het managen van wijzigingen.

Opvallend is dat ongeacht welke definitie gehanteerd wordt, men het erover eens is dat requirements management een terugkerend en complex proces is dat verweven is met het proces van Requirements Engineering. Wat logisch lijkt, het is tenslotte een management proces.

Dit onderzoek hanteert de definitie van Peatsch, Eberlein en Maurer (2003). Deze definitie geeft bondig een opsomming van de activiteiten die in vrijwel elke definitie in ieder geval genoemd worden. Voor deze definitie is gekozen omdat dit afstudeeronderzoek het begrip requirements management slechts verkent en zich richt op requirements traceability bij projecten in de theorie, praktijk en requirements traceability in een software ontwikkel- en documentatiestandaard. Het onderzoek heeft niet als doel tot een nieuwe definitie van requirements management te komen, dan wel dit gebied te onderzoeken. Deze definitie luidt: 'The goal of requirements management is to capture, store, disseminate, and manage information. Requirements management includes all activities concerned with change & version control, requirements tracing, and requirements status tracking'.

2.2 Requirements Traceability

Om een basis te vormen voor het empirisch onderzoek is de literatuur uitvoerig bestudeerd op het begrip Requirements Traceability en wat we hieronder verstaan.

Dit is geen sinecure blijkt wel. Gotel en Finkelstein beschrijven dit in 1994 al in hun 'An analysis of the requirements tracability problem'. Zij beschrijven verschillende definities die doelgericht, oplossingsgericht, informatie gedreven of in specifieke richting gemotiveerd zijn. Zij concluderen dan ook dat dit een probleem geeft omdat de scope van deze definities zeer uiteenloopt.

Om requirements traceability te realiseren wordt soms gebruik gemaakt van daarvoor bestemde tooling. Hierover zeggen Gotel en Finkelstein (1994) dat de meeste tools requirements traceability niet afdekken zoals deze wordt afgedwongen door bijvoorbeeld de DoD standard 2167A, voorloper van de Military Standard 498.

Ook wijzen zij erop dat een bepaalde vorm van requirements traceability kan ontstaan door het gebruik van talen, modellen en methoden voor ontwikkeling. Daarbij maken zij wel de kanttekening dat de kwaliteit van de uiteindelijke traceerbaarheid afhangt van de wijze waarop vooraf gespecificeerde procedures en notaties voor ontwikkeling zijn toegepast. (Gotel en Finkelstein 3.1 basic techniques)

Eberlein en Sampaio do prado leite (2002) wijzen op het belang van de link tussen requirements, de bron van requirements en code om traceability mogelijk te maken.

2.2.1 Definitie requirements traceability

Gotel en Finkelstein (1994) verwijzen naar de definitie die in de IEEE Standard 830-1984 (en later in de IEEE Standard 830-1984) wordt gehanteerd. Deze luidt "A software requirements specification is traceable if (i) the origin of each of its requirements is clear and if (ii) it facilitates the referencing of each requirement in future development or enhancement documentation" (ANSI/IEEE Standard 830-1984).

Hierbij beveelt de standaard een tweetal specifieke wijze van traceability aan, namelijk 'backward traceability' en 'forward traceability'. De standaard beschrijft dit als volgt "*Backward traceability (i.e., to previous stages of development). This depends on each requirement explicitly referencing its source in earlier documents. Forward traceability (i.e., to all documents spawned by the Software Requirements Specification). This depends upon each requirement in the SRS having a unique name or reference number.*" Ook gaat de standaard in op de letterlijke betekenis van het woord 'trace'. Hierover zegt het: "*The ability to "delineate" and "mark out" "perceptible signs of*

what has existed or happened" in the lifetime of a requirement to enable one to "pursue one's way along" this record."

Gotel en Finkelstein (1994) trekken de conclusie dat de beschrijving van de IEEE 830-1984 leidt tot de volgende definitie van **'requirements traceability'** *"refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases)."* Wat hier precies bedoeld wordt met "it's origin" geeft de standaard niet onomstotelijk aan. Wij gaan er in dit onderzoek echter vanuit dat we hiermee de herkomst bedoelen zoals een belanghebbende (stakeholder) en de oorspronkelijke bedoeling welke hij/zij hiermee gehad heeft.

De IEEE Standard 830-1984 spitst zich toe op software requirements specificaties, de IEEE Standard 610.12 – 1990 richt zich op de definities van termen gebruikt in het veld van software ontwikkeling in het algemeen.

Bij de ontwikkeling van software is in de meeste gevallen sprake van verschillende belanghebbenden met verschillende doelen en prioriteiten. Veelal worden deze belanghebbenden benoemd als stakeholders. Deze verschillende doelen en prioriteiten leiden tot verschillende requirements, echter de mogelijkheid bestaat dat de requirements van een systeem of systeemcomponenten wel degelijk een relatie met elkaar zullen hebben of juist conflicterend zijn met elkaar. Gedurende de ontwikkeling van een systeem hebben deze requirements nog steeds hun oorsprong maar maken ze ook een bepaalde ontwikkeling door. Traceability maakt het mogelijk om de oorsprong en ontwikkeling te herleiden en te beschrijven. Dit is belangrijk om requirements te begrijpen en voor verificatie. Ook kan traceability informatie geven over de rechtvaardiging, belangrijke besluiten en aannames die ten grondslag liggen aan de requirements. (Ramesh, 2001).

In dit onderzoek hanteren we de definitie van Gotel en Finkelstein gebaseerd op de IEEE Standard 830-1984. Voor deze definitie is gekozen omdat dit afstudeeronderzoek zich richt op requirements traceability bij projecten in de theorie, praktijk en requirements traceability in een software ontwikkel- en documentatiestandaard. Het onderzoek heeft niet als doel om tot een nieuwe definitie van traceability te komen dan wel dit gebied te onderzoeken.

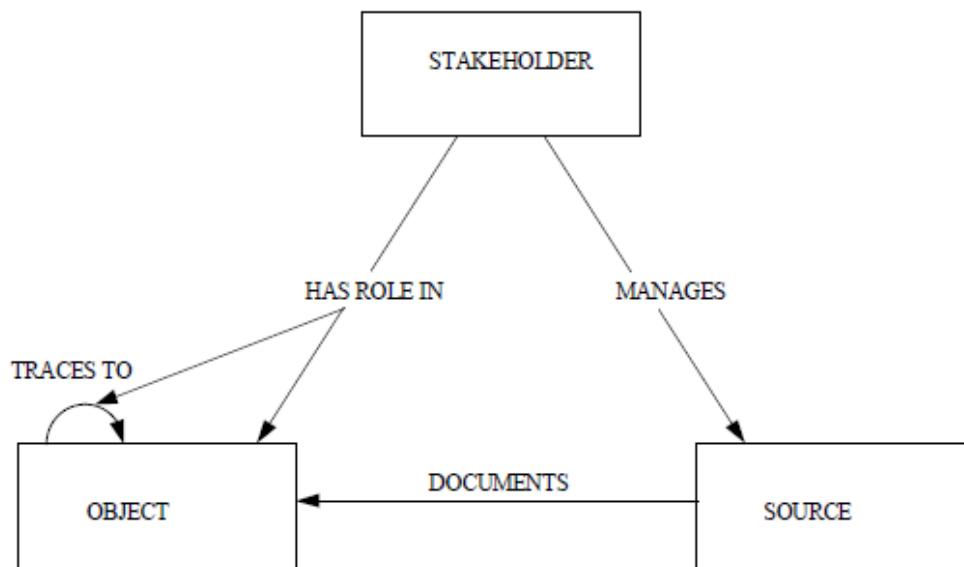
Deze definitie luidt: *"refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases)."*

2.2.2 Referentie model requirements traceability

Ramesh (2001) gaat nog verder in zijn onderzoek en ontwikkelt een aantal referentiemodellen voor requirements traceability. Het traceability meta model is een eenvoudig model wat de meest essentiële aspecten van traceability modellen beschrijft. Met dit model wordt eenvoudige informatie als het Wie, Wat, Waar, Hoe, Waarom en Wanneer afgedekt voor elke requirement. Daarmee wordt bijvoorbeeld bedoeld, 'Wat voor informatie wordt er weergegeven?', 'Wie is de Stakeholder?', Waar wordt het gepresenteerd?', 'Hoe wordt het gepresenteerd?', 'Waarom is een object gecreëerd?', Wanneer is de informatie samengesteld of aangepast?.

De drie items in het model 'OBJECT', 'STAKEHOLDER', en 'SOURCE' zijn meta classes. Het 'OBJECT' in dit model kan een requirement, aanname, ontwerp, systeemcomponent of een besluit zijn. De TRACES_TO link is een specialisatie relatie tussen deze objecten. Bijvoorbeeld een besluit kan leiden tot en bepaald ontwerp. Dit geeft dus aan 'WAT voor informatie wordt weergegeven' maar ook 'WAAROM een object gecreëerd of aangepast is.

De 'STAKEHOLDER' in dit model kan bijvoorbeeld een project manager, analist, system engineer of ontwerper zijn. Deze stakeholders spelen verschillende rollen in het vaststellen of gebruiken van de objecten zoals bijvoorbeeld een requirement of ontwerp. Het item 'STAKEHOLDER' geeft hiermee aan 'WIE de belanghebbenden zijn. De 'SOURCE' in het model bevat de traceability informatie. Dit kan bijvoorbeeld een document zijn, zoals een requirement specificatie, een fysiek medium maar ook verwijzingen naar een persoon, vergadering of telefoongesprek. Hiermee wordt aangegeven 'WAAR en 'HOE traceability informatie wordt gepresenteerd.



Figuur 2 Traceability meta model. Bron: Ramesh, B en M. Jarke (2001)

Het tweede model is wat Ramesh (2001) noemt het Low-End Use model, zie figuur 3. Dit model is wat uitgebreider en is voornamelijk gebaseerd op traceability links om onderlinge relaties tussen requirements weer te geven, requirements te koppelen aan systeemcomponenten, complianceverificatie en het beheersen van veranderingen.

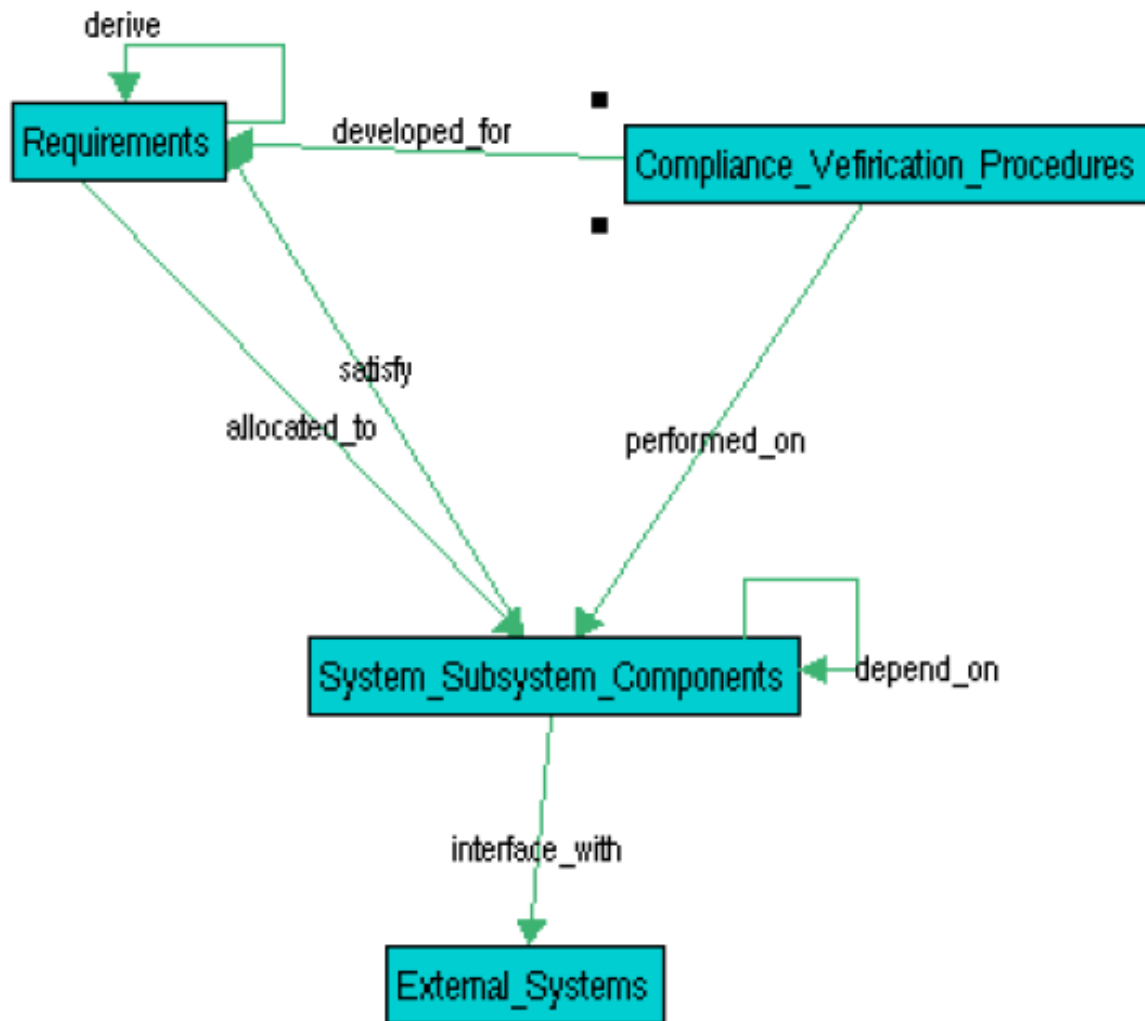
In het tweede model van figuur 3 zien we een aantal begrippen, namelijk:

- Requirements – Originele high level requirements en hiervan afgeleide requirements.
- System_Subsystem_Components – Componenten van systemen of subsystemen die invulling geven aan een requirement
- External_Systems – Externe systemen waarmee componenten een relatie kunnen hebben.
- Compliance_Verification_Procedures – Procedures om te bepalen (testen/simuleren) of aan requirements is voldaan.

Het model geeft alle objecten weer die een rol spelen bij requirements traceability. Het betreft requirements zelf en de objecten waar de requirements gedurende de life cycle mee te maken hebben.

Het model beschrijft een aantal relaties tussen de objecten, namelijk:

- 'Derive' – requirements verkregen uit requirements op een hoger niveau
- 'Developed_for' – requirements waarvoor de compliance verification procedures zijn ontwikkeld.
- 'Satisfy' – link tussen de requirements en de systeemcomponent die in deze requirements voorziet.
- 'Allocated_to' – alle requirements gerelateerd aan systeemcomponenten.
- 'Performed_on' – relatie tussen de 'compliance verification procedures' (CVP's) en de componenten, op basis waarvan geverifieerd wordt of het component voldoet aan de requirements. Feitelijk dus tests of simulaties.
- 'Depend_on' – relatie tussen systeem componenten die afhankelijk van elkaar zijn.
- 'Interface_with' – relatie om te verifiëren hoe een requirement ten aanzien van de interface met een extern systeem wordt ingevuld.



Figuur 3 Low End traceability model. Bron: Ramesh, B en Jarke (2001)

Het derde model van Ramesh en Jarke (2001) gaat nog een stuk dieper in de materie van requirements traceability en richt zich op een veel uitgebreidere toepassing van requirements traceability informatie. Dit model bestaat uit een viertal submodellen, te weten Requirements management, Design Allocation, Compliance Verification en Rationale Management. Deze modellen zijn niet opgenomen en nader toegelicht, omdat de scope van dit onderzoek is, het verkrijgen van inzicht in de verschillen tussen requirements traceability in theorie, in de Military Standard 498 en in de praktijk. Het doel is een basis of uitgangspunt te hanteren gebaseerd op de literatuur, om te kunnen bepalen hoe requirements traceability eruit moet zien. Het derde model van Ramesh en Jarke (2001) gaat verder dan dit en is te complex en gedetailleerd voor de scope van dit onderzoek.

2.2.3 Conclusie deelvraag 2

Door Gotel en Finkelstein (1994) is 19 jaar geleden al geconcludeerd dat de scope van requirements traceability zeer uiteenloopt. Deze lijkt sterk afhankelijk van de achterliggende motivatie: dient het een specifiek doel, is het oplossingsgericht of op een andere wijze specifiek gemotiveerd.

Wel blijkt als we de literatuur uitgebreid bestuderen dat een breed geaccepteerde en behoorlijk volledige definitie is: *"refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases)."* Gotel en Finkelstein (1994).

Ook biedt het tweede model van Ramesh (2001), het Low End traceability model een goede basis voor het inzicht in het begrip requirements traceability. Het omvat in ieder geval de meest essentiële aspecten van requirements traceability en beschrijft ook de relaties die een rol spelen. Op basis hiervan zou je kunnen stellen dat requirements traceability minimaal moet voldoen aan de traceability informatie:

- Wie, Wat, Waar, Hoe, Waarom en Wanneer
- Onderlinge afhankelijkheden tussen requirements
- Allocatie van requirements aan systeem componenten
- Compliance verificatie
- Wijzigingen beheer

2.3 Software ontwikkeling en documentatie standaarden

Voor het vastleggen en beheren van requirements zijn verschillende standaarden beschikbaar. Dit onderzoek richt zich op de standaard 'Military Standard 498'. Dit is de standaard die binnen organisatie onderdelen van het onderzoeksdomein ICT Services van ProRail wordt gehanteerd.

3 Military Standard 498

De Military Standard 498 is een standaard voor softwareontwikkeling en –documentatie en is ontstaan uit het samenvoegen van de standaarden Department of Defense Standard 2167A en Department of Defense Standard 7935A met een aantal verbeteringen. Één van de voornaamste hiervan is verbeterde comptabiliteit met incrementele en evolutionaire standaarden. Deze standaard beslaat het gehele proces van requirements specificaties tot aan software support en gebruikershandleidingen. Dit ziet er schematisch als volgt uit:

MIL-STD-498 Data Item Descriptions						
PLANS	CONCEPT/ REQUIREMENTS	DESIGN	QUALIFICATION TEST PRODUCTS	USER/ OPERATOR MANUALS	SUPPORT MANUALS	SOFTWARE
SDP	OCD	SSDD	STP	SUM	CPM	SPS
SIP	SSS	SDD	STD	SCOM	FSM	SVD
STRP	SRS	DBDD	STR	SIOM		
	IRS	IDD		COM		

Figuur 4 Military Standard 498 Data Item Descriptions. Bron: Norma A. Stopyra, (1996).

In deze schematische weergave zien we het proces en de producten op hoofdlijnen van het software ontwikkel proces met per onderdeel de daaraan gerelateerd Data Item Descriptions. Deze Data Item Descriptions geven een beschrijving van de inhoud en opbouw die een dergelijk document volgens de standaard moet bevatten. Het gaat om 22 Data Item Descriptions, namelijk:

- Software Development Plan (SDP)
- Software Implementation Plan (SIP)
- Software Transitie Plan (STrP)
- Operational Concept Description (OCD)
- System/Subsystem specification (SSS)
- Software Requirements Specification (SRS)
- Interface Design Description (IRS)
- System/Subsystem Design Description (SSDD)
- Software Design Description (SDD)
- Database Design Description (DBDD)
- Interface Design Description (IDD)
- Software Test Plan (STP)
- Software Test Description (STD)
- Software Test Report (STR)
- Software User Manual (SUM)
- Software Center Operator Manual (SCOM)
- Software Input/Output Manual (SIOM)
- Computer Operation Manual (COM)
- Computer Programming Manual (CPM)
- Firmware Support Manual (FSM)
- Software Product Specification (SPS)
- Software Version Description (SVD)

De standaard heeft als doel te komen tot uniforme requirements voor software ontwikkeling en documentatie. De standaard kan worden toegepast op situaties met bepaalde contractuele partijen of bij interne software ontwikkeling.

De standaard bezit een groot aantal elementen die bijdragen aan de software ontwikkeling en documentatie. Deze elementen bieden handvatten om het proces van software ontwikkeling en documenteren te ondersteunen. Afhankelijk van de situatie kunnen deze handvatten al dan niet

gebruikt worden of op de situatie toegespitst worden. Afhankelijk van het type software kan de standaard op maat toegepast worden. De verantwoordelijkheid voor het op maat toepassen van de Military Standard 498 ligt bij de partij die de behoefte aan deze software heeft. De bijlagen van de Military Standard 498 bieden enkele concrete voorbeelden van hoe de standaard op maat gemaakt kan worden afhankelijk van het software type en de strategie. In figuur 5 zien we een drietal strategieën namelijk 'Grand design', 'Incremental' en 'Evolutionary'. De 'Grand design' is een strategie waarin je alles in één keer doet en elke stap maar één keer zet. De 'Incremental' strategie brengt in kaart wat de gebruikers wensen zijn en wat de systeemvereisten zijn. Vervolgens verloopt alles in deze strategie in delen, eerst één build waarin een gedeelte wordt gerealiseerd en dan de volgende build. De 'Evolutionary' strategie werkt op vergelijkbare wijze maar met het verschil dat geaccepteerd wordt dat niet alle requirements van tevoren bekend zijn. Deze verschillen worden ook aangegeven in de tabel onderaan in figuur 5.

G.3 **Candidate program strategies.** DODI 8120.2 describes three basic program strategies plus a generic strategy called "other," encompassing variations, combinations, and alternatives to the three. DODI 5000.2 identifies similar strategies, called acquisition strategies. The three basic strategies are summarized below and in Figure 7.

- Grand design.** The "grand design" strategy (not named in DODI 5000.2 but treated as one strategy) is essentially a "once-through, do-each-step-once" strategy. Simplistically: determine user needs, define requirements, design the system, implement the system, test, fix, and deliver.
- Incremental.** The "incremental" strategy (called "Preplanned Product Improvement" in DODI 5000.2) determines user needs and defines the system requirements, then performs the rest of the development in a sequence of builds. The first build incorporates part of the planned capabilities, the next build adds more capabilities, and so on, until the system is complete.
- Evolutionary.** The "evolutionary" strategy (called "evolutionary" in both DOD Instructions) also develops a system in builds, but differs from the incremental strategy in acknowledging that the user need is not fully understood and all requirements cannot be defined up front. In this strategy, user needs and system requirements are partially defined up front, then are refined in each succeeding build.

Program Strategy	Define All Requirements First?	Multiple Development Cycles?	Field Interim Software?
Grand Design	Yes	No	No
Incremental (Preplanned Product Improvement)	Yes	Yes	Maybe
Evolutionary	No	Yes	Yes

FIGURE 7. Key features of three DOD program strategies.

Figuur 5 Key features of three DOD program strategies. Bron: Military Standard 498, <http://www.everyspec.com>

Die elementen die de standaard biedt bestaan uit een beschrijving van de scope van de Military Standard 498 en het benoemen van uniforme begrippen zoals 'acquirer', 'contract' of 'Statement of Work'. Deze begrippen krijgen een vaste definitie vanuit de standaard, bijvoorbeeld 'Statement of Work.' Dat is de lijst met taken die uitgevoerd moeten worden door de ontwikkelaar. Vervolgens beschrijft de Military Standard 498 een aantal algemene requirements. De standaard beschrijft dat er een software ontwikkel proces vastgesteld moet worden, beschreven moet worden in een software ontwikkel plan en geeft uitvoerig aan wat het hierin verwacht. De onderdelen die bijdragen aan requirements traceability worden uiteengezet in paragraaf 3.1

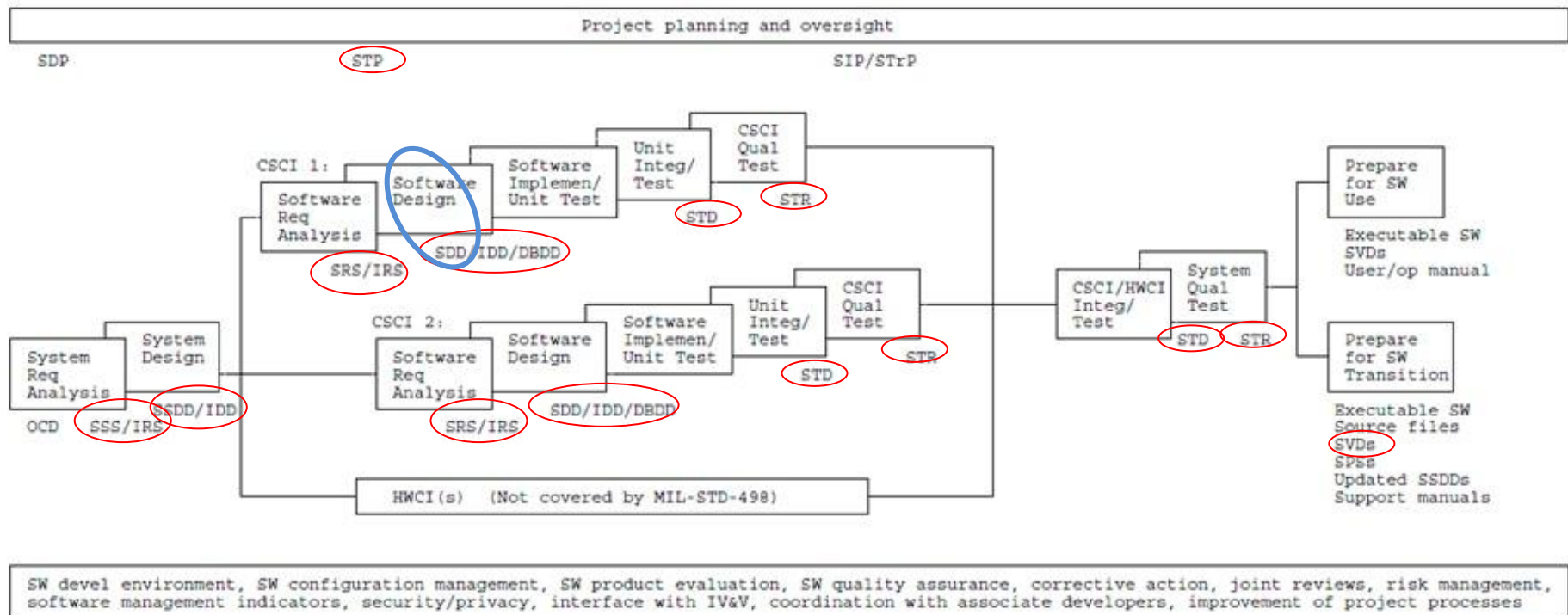
De ontwikkelaar dient eveneens systematisch gedocumenteerde methodes toe te passen in de software ontwikkelactiviteiten en hieraan te refereren of deze te beschrijven in dit software ontwikkel plan. Hetzelfde geldt voor het beschrijven van requirements, ontwerp, code, test procedures, test plan en test resultaten. Er worden ook specifieke eisen gesteld aan hoe de ontwikkelaar om dient te gaan met kritische requirements zoals bijvoorbeeld 'Safety assurance' of 'Privacy assurance'. In het geval van de 'Privacy assurance' wordt bijvoorbeeld gesteld dat de ontwikkelaar moet vaststellen of er

sprake is van privacy gevoelige aspecten of systemen. Indien dit het geval is dient de ontwikkelaar zorg te dragen voor een strategie om de privacy te waarborgen.

Kortom, de Military Standard 498 bevat een aantal elementen die feitelijk beschrijven hoe het ontwikkelproces er volgens de standaard uit moet zien. Dit wordt ondersteunt met gedetailleerde beschrijvingen per vereiste en 22 hieraan gekoppelde Data Item Descriptions. In figuur 6 is een voorbeeld weergegeven hoe dit er uit zou zien bij 1 build. Met 1 build wordt bedoeld één cycles waarin de stappen van specificeren, ontwerpen, bouwen en testen voor een computer software configuration item worden voltooid. Uiteraard kan er sprake zijn van meerdere computer software configuration items en daarmee dus van meerdere builds.

Het biedt deze voorbeelden ook met meer Iteratie of voor Reengineering van software. In figuur 6 zien we (met afkortingen) alle 22 Data Item Descriptions benoemd en gekoppeld aan de proces stappen die de Military Standard 498 voorschrijft. In de Data Item Descriptions die rood omcirkelt zijn is een hoofdstuk requirements traceability opgenomen en beschrijft de standaard welke traceability dit hoofdstuk moet bieden.

Deze Data Item Descriptions zijn een leidraad. In welke mate deze Data Item Descriptions opgevolgd moeten worden hangt af van het type project. De standaard legt niets op maar is bedoeld als hulpmiddel om op uniforme wijze requirements vast te leggen en te voorzien in de noodzakelijk documentatie van een software systeem. De standaard beschrijft vooral ook 'wat' te doen en niet 'hoe' dit gedaan moet worden. Keuzes voor bijvoorbeeld modellering technieken maakt de standaard niet.



Note: All activities may be more ongoing, overlapping, and iterative than the figure is able to show.

Figuur 6 Weergave ontwikkel proces Military Standard 498. Bron: Abelia.com

3.1 Requirements traceability Military Standard 498

In de standaard wordt op verschillende wijze aandacht besteed aan Traceability. De standaard geeft in vrijwel elk onderdeel aandacht aan het belang van traceability. In paragraaf 5.4.2 van de Military Standard 498 (1994) wordt nadrukkelijk de verantwoordelijkheid bij de ontwikkelaar gelegd om zorg te dragen voor traceability tussen de systeem componenten en de systeem requirements. Daarbij geeft het aan dat alle items die van toepassing zijn opgenomen dienen te worden in de traceability secties van de System/Subsystem Design Description (SSDD). De Data Item Description geeft aan weke secties dit zijn.

Paragraaf 5.5 van de Military Standard 498 (1994) beschrijft hetzelfde voor de traceability tussen de Computer Software Configuration Item (CSCI) requirements en de systeem requirements. Hierbij wordt verwezen naar de traceability sectie in de Software Requirements Specification (SRS) Data Item Description.

Paragraaf 5.6.2 van de Military Standard 498 (1994) beschrijft de verantwoordelijkheid voor de ontwikkelaar om zorg te dragen voor traceability tussen de software units en de Computer Software Configuration Item requirements. Hierbij wordt verwezen naar de traceability sectie in de Software Design Description (SDD) Data Item Description. Daarbij is het van belang dat in het resultaat alle van toepassing zijnde onderdelen uit het architectuur ontwerp terug komen als ook de traceability secties van de Software Design Description. Dit is feitelijk de relatie tussen de processtap die wordt genomen namelijk 'Software design' en de Software Design Description. Zie de blauwe cirkel in figuur 6.

In paragraaf 5.9.3, 5.11.3, 5.13.4, van de Military Standard 498 (1994) wordt gewezen op de verantwoordelijkheid van de ontwikkelaar om zorg te dragen voor traceability tussen testcases en respectievelijk Computer Software Configuration Item requirements, system requirements, 'Computer Software Configuration Item source files and software units' en tussen hardware specificaties en de requirements hiervan.

In appendix D wordt nogmaals benadrukt dat een software product pas een gegeven set van items afdekt als deze allemaal verwerkt zijn in het software product. Hierbij wordt gewezen op de neerwaartse traceability van requirements, ontwerp en testplanning zoals dit in de Data Item Descriptions van de standaard is beschreven.

In een groot deel van de Data Item Descriptions (DID's) wordt aandacht besteed aan de verschillende soorten traceability. Het begint met de concept/requirements documentatie. In de System/Subsystem Specification hoofdstuk 3 wordt aangegeven dat het van belang is voor het testen en de traceability dat iedere requirement van een unieke project identifier wordt voorzien. Deze project identifier is traceerbaar en daarmee de hierdoor geïdentificeerde requirements eveneens. Er wordt geen verdere systematiek benoemd ten aanzien van deze identifier. In hoofdstuk 5 van deze specificatie dient te worden aangegeven welke subsystem requirement geadresseerd is aan welke system requirement. Dit geldt ook vice versa voor system requirements die gekoppeld zijn aan een subsystem die in deze specificatie behandeld wordt. Daarbij tevens verwijzend naar eventuele Interface Requirements Specificaties waarmee een relatie is. In dat geval dient gerefereerd te worden aan die Interface Requirement Specificatie.

De Data Item Description van de Software Requirements Specification besteed eveneens in hoofdstuk 3 en hoofdstuk 5 aandacht aan traceability. Ook voor veel overige onderdelen zoals de Interface Design Description of Database Design Description beschrijft de standaard welke traceability opgenomen dient te worden.

Gotel en Finkelstein (1995) beschrijven twee 'zorgen' die het risico weergeven van het op deze wijze om gaan met traceability. Bij het gebruik van de standaard is men tenslotte vrij om al dan niet invulling te geven aan de traceability die wordt beschreven in de Data Item Descriptions. Dit kan er dus toe leiden dat men dit niet doet. Uiteraard ontstaat er dan ook geen traceability.

3.2 Data Item Descriptions in relatie tot de standaard

In figuur 4 hebben we een schematische weergave gezien van de Data Item Descriptions. De Data Item Descriptions zijn als bijlage opgenomen in de Military Standard 498. Zoals we in voorgaande paragraaf 3.1 al hebben gezien wordt in de de hoofdtekst van de Military Standard 498 verwezen naar de Data Item Descriptions.

Met hoofdtekst bedoelen we de hoofdstukken van de Military Standard 498. Dus geen bijlage of andere gerelateerde documenten. Als we dit nader bestuderen zien we dat de directe relatie tussen de hoofdtekst en de Data Item Descriptions gelegd wordt door Hoofdstuk 5 van de Military Standard 498. In hoofdstuk 5 worden de activiteiten benoemd die volgens de standaard uitgevoerd moeten worden gedurende het ontwikkelproces, zie figuur 7.

Activity
5.1 Project planning and oversight
5.2 Establishing a software development environment
5.3 System requirements analysis
5.4 System design
5.5 Software requirements analysis
5.6 Software design
5.7 Software implementation and unit testing
5.8 Unit integration and testing
5.9 CSCI qualification testing
5.10 CSCI/HWCI integration and testing
5.11 System qualification testing
5.12 Preparing for software use
5.13 Preparing for software transition
Integral processes:
5.14 Software configuration management
5.15 Software product evaluation
5.16 Software quality assurance
5.17 Corrective action
5.18 Joint technical and management reviews
5.19 Other activities

Figuur 7 'Military Standard 498 Software Development Activities'. Bron: Military Standard 498, <http://www.everyspec.com>

Aan elke activiteit is een paragraaf gewijd. In deze paragraaf en subparagrafen wordt besproken wat deze activiteit moet opleveren. Daarbij wordt in de meeste paragrafen verwezen naar een Data Item Description, zie figuur 8. De resultaten van de activiteit moeten aan alle relevante items in het Data Item Description voldoen. De Data Item Descriptions vormen formeel de requirements voor de data die door de leverende partij opgeleverd moeten worden.

Onderstaande afbeelding geeft de relaties tussen paragrafen en Data Item Description:

Reference Para	DID Number	DID Title
5.1.1	DI-IPSC-81427	Software Development Plan (SDP)
5.1.2, 5.1.3	DI-IPSC-81438	Software Test Plan (STP)
5.1.4	DI-IPSC-81428	Software Installation Plan (SIP)
5.1.5	DI-IPSC-81429	Software Transition Plan (STrP)
5.3.2	DI-IPSC-81430	Operational Concept Description (OCD)
5.3.3	DI-IPSC-81431	System/Subsystem Specification (SSS)
5.3.3, 5.5	DI-IPSC-81434	Interface Requirements Specification (IRS)
5.4.1, 5.4.2, 5.13.5	DI-IPSC-81432	System/Subsystem Design Description (SSDD)
5.4.1, 5.4.2, 5.6.1, 5.6.2, 5.6.3	DI-IPSC-81436	Interface Design Description (IDD)
5.5	DI-IPSC-81433	Software Requirements Specification (SRS)
5.6.1, 5.6.2, 5.6.3	DI-IPSC-81435	Software Design Description (SDD)
5.4.1, 5.6.1, 5.6.3	DI-IPSC-81437	Database Design Description (DBDD)
5.9.3, 5.11.3	DI-IPSC-81439	Software Test Description (STD)
5.9.7, 5.11.7	DI-IPSC-81440	Software Test Report (STR)
5.12.1, 5.13.1, 5.13.2, 5.13.4	DI-IPSC-81441	Software Product Specification (SPS)
5.12.2, 5.13.3	DI-IPSC-81442	Software Version Description (SVD)
5.12.3.1	DI-IPSC-81443	Software User Manual (SUM)
5.12.3.2	DI-IPSC-81445	Software Input/Output Manual (SIOM)
5.12.3.3	DI-IPSC-81444	Software Center Operator Manual (SCOM)
5.12.3.4	DI-IPSC-81446	Computer Operation Manual (COM)
5.13.6.1	DI-IPSC-81447	Computer Programming Manual (CPM)
5.13.6.2	DI-IPSC-81448	Firmware Support Manual (FSM)

Figuur 8 'Military Standard 498 Data requirements'. Bron: Military Standard 498, <http://www.everyspec.com>

Als voorbeeld nemen we de activiteit 'Software requirements analysis', zie figuur 7. In de betreffende paragraaf 5.3 treffen we de vereiste dat de ontwikkelaar moet bijdragen aan het operationele concept voor het systeem. Dat houdt ook in zoals expliciet wordt aangegeven dat het resultaat alle relevante onderdelen van het Data Item Description Operational Concept Description bevat.

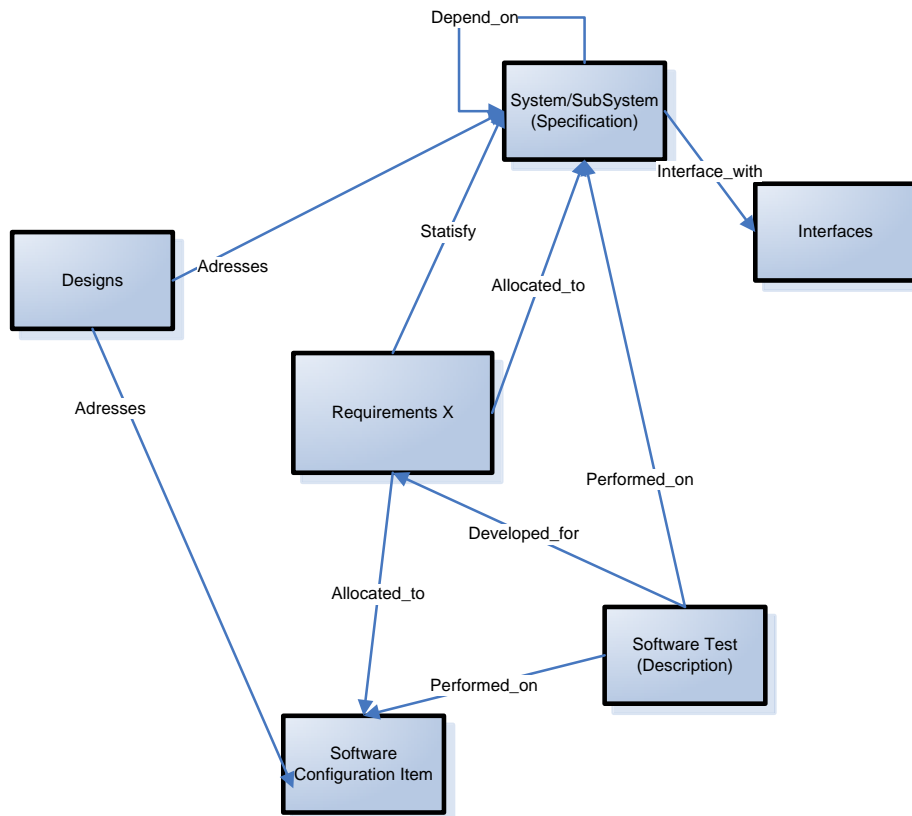
3.3 Conclusie deelvraag 3

De Military Standard 498 is een standaard voor software ontwikkeling en documentatie van planfase tot oplevering. De standaard biedt een helder model van alle documentatie die het vereist. Per document in dit model biedt de standaard databeschrijvingen per document.

De middelen die het toepast om deze requirements traceability te behalen zijn:

- De standaard legt nadrukkelijk de verantwoordelijkheid voor requirements traceability mede bij de ontwikkelaar. Niet alleen met betrekking tot specificatie en ontwerp maar ook voor test cases, CSCI requirements, system requirements, 'CSCI source files and software units.
- De standaard vereist identificatie voor requirements.'
- De standaard vereist dat alle relevante items opgenomen worden in de daartoe bestemde secties van de documentatie ten behoeven van requirements traceability. De standaard benoemt dit in het algemeen maar ook expliciet voor de onderdelen:
 - o System/Subsystem specification (SSS)
 - o System/Subsystem Design Description (SSDD)
 - o Software Requirements Specification (SRS)
 - o Interface Requirements Specification (IRS)
 - o Software Design Description (SDD)
 - o Interface Design Description (IDD)
 - o Database Design Description (DBDD)

Kortom de standaard geeft een model en structuur ten aanzien van het ontwikkelproces met een documentatieset en schrijft in verschillende hoofdstukken, paragrafen en Data Item Descriptions voor hoe requirements traceability te borgen. De requirements traceability die de Military Standard 498 biedt is weergegeven in onderstaand conceptueel model met bijbehorende relaties. Het model laat duidelijk overeenkomsten zien met de literatuur. Het model van Ramesh en Jarke (2001) zie figuur 3, maakt eveneens onderscheid in systemen en subsystemen, relaties met externe systemen en test specificaties om te verifiëren of aan de requirements voldaan is. Ook leggen zij vergelijkbare relaties. Bijvoorbeeld de relaties 'Allocated_to' of 'Depend_on' komen in beide modellen terug.



Figuur 9 'Conceptueel model Requirements Traceability Military Standard 498'

Toelichting relaties:

- *Satisfy* – Requirements waarin wordt voorzien door een systeem of subsystem.
- *Allocated_to* - Traceability van elk System requirement die toegewezen is aan die specifieke Subsystem Specification naar dat specifieke Subsystem requirement en aan Software Configuration item.
- *Depend_on* – Subsystem requirement naar het systeem welke het adresseert.
- *Interface_with* – Requirements betreffende een specifiek Subsystem.
- *Developed_for* – Kwalitatieve methodes beschreven om te verifiëren of aan de requirements is voldaan. (testing/inspection)
- *Performed_on* – Traceability van elke test case in de STD naar het Systeem of het Computer Software Configuration Item requirements die het betreft. Traceability van elke Systeem of Computer Software Configuration Item welke is opgenomen in een specifieke STD.
- *Adresses* – Requirements traceability tussen de ontwerpen (database, software, interface) en de systemen en software configuration items die het systeem adresseert
- *Interface_with* – Traceability van de interface specificatie naar het betreffende System of Subsystem.

Ter verduidelijking van het conceptuele model en zijn relaties kunnen we een voorbeeld geven. Stel we ontwikkelen een systeem voor Enterprise project portfolio management. Dit systeem stelt high level requirements waaronder bijvoorbeeld:

- 1) Het moet mogelijk zijn programma's te administreren.
- 2) Programma's worden voorzien van een unieke identificatie.
- 3) Het moet mogelijk zijn om projecten te administreren.
- 4) Projecten worden voorzien van een unieke identificatie. Indien het project onderdeel is van een programma zal de identificatie als prefix de identificatie van het programma hebben.
- 5) Het moet mogelijk zijn om project-portfolio's te administreren.
- 6) Programma's kunnen uit meerdere projecten bestaan.
- 7) Er moet een koppeling zijn met het financiële administratie systeem.

In dit voorbeeld zullen de requirements mogelijk leiden tot een drietal subsystemen, namelijk een subsysteem voor het administreren van de programma's, een subsysteem voor het administreren van portfolios en een subsysteem voor het administreren van projecten. Het subsysteem voor het administreren van programma's zal dus invulling geven aan de eerste twee systeem requirements. De relatie 'Allocated to' betreft in deze de relatie tussen de requirement om programma's te kunnen administreren en de subsystem requirement dat een programma een unieke identificatie dient te hebben.

De subsystemen zullen ontworpen worden. In het ontwerp willen we terug zien dat de samenstelling van de identificatie van een project o.a. tot stand is gekomen door requirement nummer vier. Ook willen we in de testbeschrijving een testscenario terug zien waarin een project onderdeel uit maakt van een programma en we controleren of dit project ook de juiste identificatie krijgt zoals vereist in requirement nummer vier.

Requirement nummer zeven vereist een koppeling met het financiële informatie systeem. Voor deze requirement zal een interface ontwikkelt moeten worden. Deze interface zal dus een koppeling met het systeem verzorgen en daarmee invulling geven aan requirement zeven.

4 Analyse van requirements traceability in projecten

In dit hoofdstuk wordt een analyse gegeven van verschillende projecten in het domein van ICT Services bij ProRail. Per project is geanalyseerd of en hoe requirements traceability aanwezig is. Er zijn daarbij zowel projecten bekeken waarbij geen standaard is toegepast als projecten waarbij de Military Standard 498 is toegepast. Van ieder project is een tweetal requirements getraceerd zowel backwards als forwards traceability. In figuur 10 treffen we een overzicht van de projecten. Daarin is te zien of deze projecten de Military Standaard hebben toegepast en de requirements die getraceerd zijn.

Project	Gebruik Military Standard 498	Requirements die getraceerd zijn
ASTRIS	Ja	<ol style="list-style-type: none"> 1) Systeem requirement UC-VHR <i>"Astris moet aan bedienapplicaties de mogelijkheid bieden om (meerdere) Trdl-verhinderingen voor rijweginstelling aan te brengen op seinen, secties, wissels en kruisingen. 235"</i> 2) Systeem requirement <i>"De on-line middelen voor beheer (zie o.a. par. 3.13.2) dienen gebruikt te kunnen worden 3040 zonder dat de werking van Astris buiten specificaties treedt. Dit geldt dus ook voor zaken als 3041 eventmeldingen en logging."</i>
BBMS	Nee	<ol style="list-style-type: none"> 1) Requirement 2.1.3 <i>"Daar het contract kan bestaan uit meerdere metingen per jaar, wordt op de dekkingslaag contract de frequentie aangegeven. Voor de dekkingslagen planning, uitvoering en meetgegevens zijn per laag meerdere dekkingen per jaar beschikbaar."</i> 2) Requirement
SBG++	Ja	<ol style="list-style-type: none"> 1) US039 <i>"SBG++ toont een grafiek met daarin alle relevante sporen naast elkaar. Deze sporen worden weergegeven als verticale lijnen. In deze grafiek loopt de tijd van beneden naar boven"</i>. 2) US245 <i>"SBG++ moet in staat zijn om berichtverlies op de ARP interface te detecteren. [IRS-MARC] definieert al een mechanisme dat hierin kan helpen: elk bericht heeft een volgnummer. Met behulp van dit volgnummer kan eenvoudig worden bepaald of er berichten zijn gemist"</i>.
Routelint	Ja	<ol style="list-style-type: none"> 1) RMR.CD.2 <i>"Eigen trein hinderende andere trein"</i>. 2) RMR.TST.6 <i>"Falen van toestandsovergangen"</i>.
RRCB	Nee	<ol style="list-style-type: none"> 1) High Level Requirement 1. <i>"Vastleggen PGO contractafspraken"</i>. 2) High Level Requirement 14. <i>"Afwijking met geaccordeerde voostel moeten binnen de oplosperiode afgerond zijn. Het systeem bewaakt dit."</i>

Figuur 10 'Overzicht van projecten, toepassing Military Standard 498 en getraceerde requirements'.

Voor ieder project wordt een korte inleiding gegeven om een idee te geven van het soort project en het domein. Daarop volgt een korte beschrijving van de structuur waarin de requirements zijn vastgelegd. Vervolgens zal de requirements traceability van een tweetal requirements uit het betreffende project worden weergegeven. Het traceren van de requirements is uitgevoerd door in de requirements specificatie of, indien aanwezig, in de traceability matrix de identificatie of prefix van de requirement te achterhalen. Daarop is met behulp van deze identificatie of prefix en de 'find' functionaliteit in de projectdocumentatie getracht de betreffende requirement terug te vinden.

4.1 Project en domein ASTRIS

Applicatie: ASTRIS

Bedrijfsprobleem dat door de applicatie ondersteund wordt.

De TreinBeheersingsSystemen (TBSn) zijn een vitale schakel in de treinbesturings- en treinbeveiligingsketen, zonder welke er geen treinverkeer mogelijk is. ASTRIS (Aansturing en Status melding van de RailInfraStructuur) is een nieuwbouw product, welke de vervanger is van het huidige conglomeraat van TreinBeheersingsSystemen. In het voorjaar van 2013 zal ASTRIS worden beproefd. De TreinBeheersingsSystemen bevinden zich hiërarchisch gezien boven de beveiligingssystemen, ze sturen de beveiligingssystemen aan, en onder de treinbesturingsystemen, waar ze bedienopdrachten van ontvangen en status aanmelden.

De beveiligingssystemen zijn de autonome fail-safe systemen die zorg dragen voor ontsporingsbeveiliging en treinseparatie en voeren sturingen alleen uit als dat veilig kan.

De doelstellingen van ASTRIS zijn:

- Het kunnen bedienen van de infrastructuur op een voor de bedienende applicatie (en daarmee voor de treindienstleider) uniforme wijze, ongeacht het onderliggende beveiligingssysteem.
- Het melden van de actuele toestand van de railinfrastructuur aan afnemers (treindienstleider, treinvolgsysteem).
- Het veilig kunnen overdragen van bedienbevoegdheden naar lokale bedienaars (b.v. rangeerders, werkploegen, brugwachters, etc.).

Korte omschrijving van de functionaliteiten

Concreet omvat Astris de volgende functionaliteit:

- het routeren van opdrachten van de bedienapplicaties naar de beveiligingen
- het routeren van informatie van de onderliggende beveiligingen naar de afnemende applicaties
- controles m.b.t. bevoegdheden
- controles om conflicterend infragebruik te voorkomen
- controles om opdrachten te weigeren die met de actuele toestand van de beveiliging en/of Astris niet mogelijk zijn
- treindienstleider-verhinderingen-functionaliiteit (zowel aanbrengen/verwijderen als controle bij opdrachten). Deze functionaliteiten stellen de treindienstleider in staat bepaalde verhinderingen aan te brengen op infrastructurele elementen (sein, wissel etc.).
- het bijhouden en beschikbaar stellen van de actuele toestand en het gebruik van de railinfrastructuur.

Gezien de functionaliteit heeft ASTRIS een zeer hoog vereiste beschikbaarheid en betrouwbaarheid. ASTRIS voldoet aan de CENELEC SIL1 norm (Cenelec 2011). De Cenelec norm is een Europese Standard welke het proces en de technische requirements specificeert voor de ontwikkeling van software in programmeerbare elektronische spoorwegbeheer en veiligheidssystemen. Het is er op gericht om toe te passen in systemen waar veiligheid een belangrijke rol speelt. ASTRIS voldoet dus aan de vereisten van deze Europese Standaard op het niveau SIL1.

4.1.1 Requirements structuur ASTRIS

In het project ASTRIS is duidelijk sprake van een eenduidige structuur waarmee requirements worden vastgelegd. Requirements worden zeer nauwkeurig en gedetailleerd vastgelegd volgens een vooraf overeen gekomen werkwijze binnen het project. Er is een overkoepelend document dat het organisatieprobleem, de systeemarchitectuur/technische beschrijving en een leeswijzer beschrijft.

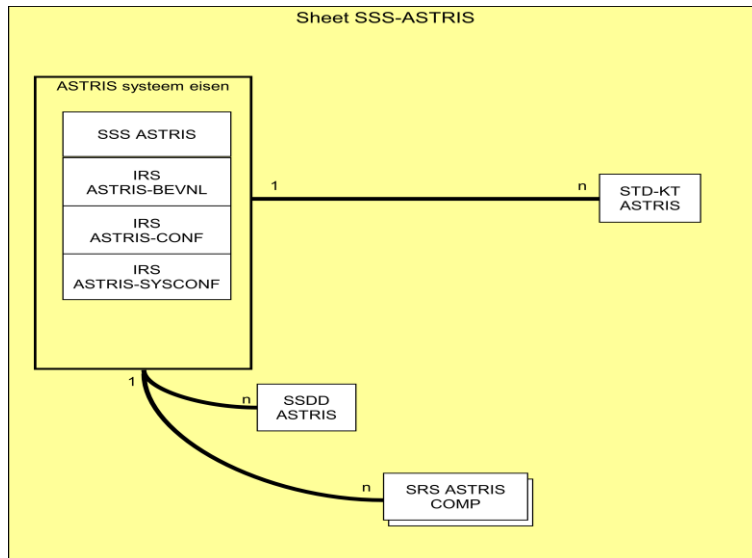
Requirements documentatie in het project is vastgelegd volgens de software ontwikkel- en documentatiestandaard Military Standard 498. Daarbij is de documentatie opgezet volgens de Data Item Descriptions van de Military Standard 498. Dit zijn documenten als een System/Subsystem Specification, Software Requirements Specification, System/Subsystem Design Specification. In bijlage I zijn enkele voorbeelden van deze documentatie uit het project ASTRIS opgenomen.

4.1.2 Requirements traceability project ASTRIS

Binnen het project is voorzien in de traceability van requirements. Er is gewerkt met een traceability matrix in Excel. Deze is opgedeeld naar systeemeisen en componenten. In deze sheet is ook

document- en versiebeheer toegepast. Je ziet bijvoorbeeld dat een Software Requirements Specificatie gerelateerd kan zijn aan meerdere Software Test Descriptions of Software Design Descriptions.

In het eerste tabblad wordt schematisch weergegeven welke relaties gebruikt worden voor de traceability.



Figuur 11 'Schematische weergave traceability relaties ASTRIS systeemeisen'. Bron: ASTRIS-traceabilitymatrix.

ASTRIS component eisen

In onderstaand figuur zijn de documenten en de relatie weergegeven die gebruikt worden voor deze traceability.

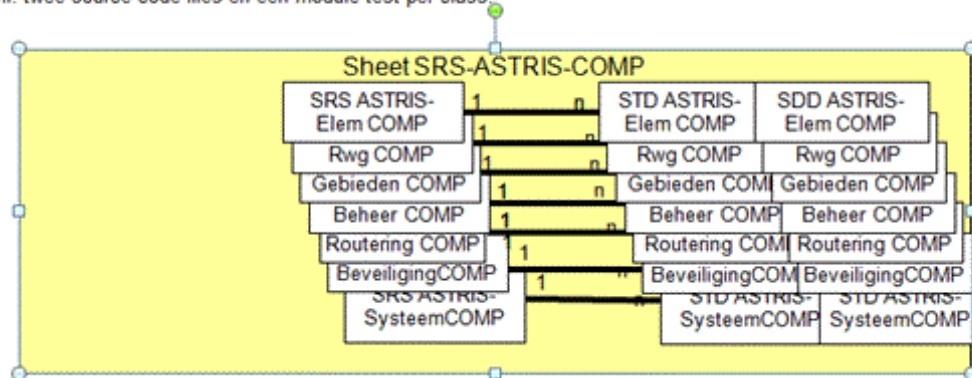
De traceability van de ASTRIS systeem eisen zijn uitgewerkt in de sheet "ASTRIS component eisen"

De traceability van de ASTRIS component eisen (SRS) naar ASTRIS component ontwerp (SDD) zijn uitgewerkt in de sheet "ASTRIS component eisen"

In deze sheet worden de volgende koppelingen gelegd:

- SRS eis <-> ontwerp (ASTRIS-XXX-SDD)
- SRS eis <-> testgevallen (STD ASTRIS-XXX COMP)

Verdere tracing naar source code en component (module) test volgt één-op-één uit de class-definities uit de SDD's, nl. twee source code files en één module test per class.



Figuur 12 'Relaties ASTRIS systeemeisen'. Bron: ASTRIS-traceabilitymatrix.

Ook wordt uitgebreid beschreven welke informatie een kolom bevat. Dit per tabblad van de matrix.

E	F	G
Sheet	Uitleg	
Inleiding	Bevat de inleiding en uitleg van de ASTRIS traceability matrix	
Documenten	Bevat informatie over de documenten waarin de eisen / testgevallen / ... staan die gebruikt zijn in de traceability matrix.	
ASTRIS systeem eisen met kolom:	Traceability matrix van het ASTRIS systeem.	
SSS / IRS Eis	n.v.t.	
Eis vervallen [[ja/]	Geeft aan of een eis vervallen is.	
ISO 9126	ISO 9126 bevat kwaliteitseigenschappen van softwareproducten. Deze attributen worden binnen de testaanpak van ASTRIS gebruikt. Elke eis is gekoppeld aan een kwaliteitsattribuut.	
Safety relevant [[ja/nee]	Geeft aan of een eis relevant is m.b.t. safety. Voor externe interfaces is dit nog niet bepaald, maar is hier een Application Condition ingevuld die in de GASC terugkomt.	
Hazard	Indien safety-relevant = J: het hazard dat deze eis betreft.	
SSDD ASTRIS	Geeft aan waar de eis volgens het SSDD ASTRIS geïmplementeerd moet worden.	
Kwalificatie voorziening	Invulling van hoofdstuk 4 van de SSS (zie MIL-STD-498, SSS-DID)	
Demonstratie (D)	(zie MIL-STD-498, SSS-DID)	
Test (T)	(zie MIL-STD-498, SSS-DID)	
Analyse (A)	(zie MIL-STD-498, SSS-DID)	
Inspectie (I)	(zie MIL-STD-498, SSS-DID)	
Speciale methode (S)	(zie MIL-STD-498, SSS-DID)	
Controle kwa. Voorz.	(hidden) om dmv kleur snel te kunnen zien of kwal voorz. Ingevuld is.	
Opmerking	indien geen testgeval voor deze eis: hoe dan wel te verifiëren.	
STD ASTRIS Testgeval	n.v.t.	
SRS Eis identificatie	concat van volgende kolommen. Is rood als er geen tracement is.	
Niet-SRS-tracement	als een systeem-eis geen onderliggende SRS-eis heeft staat hier hoe deze eis wel afgedekt is.	
SRS Elementen	SRS ELEMENT eis die traceert naar de SSS / IRS eis.	
SRS Rijkwegen	SRS RIJWEG eis die traceert naar de SSS / IRS eis.	
SRS Gebieden	SRS GEBIED eis die traceert naar de SSS / IRS eis.	
SRS Beheer	SRS BEHEER eis die traceert naar de SSS / IRS eis.	
SRS Beveiliging	SRS BEVEILIGING eis die traceert naar de SSS / IRS eis.	
SRS Routing	SRS ROUTERING eis die traceert naar de SSS / IRS eis.	
SRS Systeem	SRS SYSTEEM eis die traceert naar de SSS / IRS eis.	

Figuur 13 'Inleiding en uitleg traceability matrix' Bron: ASTRIS-traceabilitymatrix.

Figuur 14 toont de traceability matrix. Hierin is te zien dat de systeem-eis 'ASTRIS-UC-VHR' vanuit de System/Subsystem Specification (SSS) en/of de Interface Requirements Specification getraceerd worden naar de betreffende requirement specificatie in de Software Requirements Specification en naar de betreffende Software Test Description. Zoals te zien in het voorbeeld is deze systeem-eis gerelateerd aan meerdere software requirement specificaties.

De opbouw van de identificatie is als volgt. Uiteraard is ASTRIS de projectnaam, UC staat voor Use Case en VHR voor 'verhinderende rijweginstelling'. De Software eisen waarnaar verwezen wordt volgen een opbouw in de identificatie die is beschreven in de traceability matrix. EC staat in dit voorbeeld voor Element Component en EMC voor Element Manager Component. Alle drie de eisen bevinden zich dan ook in de Software Requirements Specificatie die als identificatie in de naam ELEMENTCOMP draagt namelijk het document ASTRIS-ELEMENTCOMP-SRS. De nummers in de identificatie zijn opeenvolgende nummers binnen de betreffende prefix. Deze relatie tussen de componenten en naamgeving is opgenomen in de inleiding van de traceability matrix. Op deze manier is dus te volgen in welke SRS deze software eisen terug komen.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	SSS/IRS Eis identificatie	Eis vervalst (jaar)	ISO 9126 kwaliteitsattrib	Safety relevant (ja/nee)	Hazard	SSDO ASTRIS component	Kwalificatie voorziening					Controle	Opmerking Verificatiewijze	STD ASTRIS KT/T Testgev.	SRS Eis identificatie
							O	T	A	I	S				
1	ASTRIS-UC-VHR		Geschiedheid	N										ASTRIS-TG-VHR*	EMC-IF-03, EMC-FUNC-004, EC-DATA-07
2	ASTRIS-UC-VHR		Geschiedheid	N										ASTRIS-TG-VHR*	EMC-IF-07, EMC-FUNC-002, GMC-EXTIF-20, GMC-FUNC-097
3	ASTRIS-UC-TSR		Geschiedheid	N										ASTRIS-TG-TSR*	EMC-IF-10, EMC-FUNC-005
4	ASTRIS-UC-TSB		Geschiedheid	N										ASTRIS-TG-TSB*	EMC-IF-08, EMC-FUNC-003, GMC-EXTIF-21, GMC-FUNC-098
5	ASTRIS-UC-BE		Geschiedheid	N										ASTRIS-TG-BE*	EMC-IF-06, EMC-FUNC-006
6	ASTRIS-UC-ROUTE		Geschiedheid	N										ASTRIS-UC-ROUTE	EMC-IF-004, EMC-IF-005, EMC-IF-006

Figuur 14 'Traceability systeem eisen'. Bron: ASTRIS-traceabilitymatrix.

In Figuur 15 zien we de systeem eis 'ASTRIS-UC-VHR' in de System/Subsystem Specification (SSS). Hierop openen we de Software Requirements Specification 'ASTRIS-UC-VHR' en de Software Test Description 'ASTRIS-STD-KT'. In figuur 16 zien we een onderdeel van het testgeval ASTRIS-TG-VHR. Het testgeval is uitgebreider. Zoals te zien is, is het opgesplitst en bevat het verschillende testcases.

<u>Trdl-verhinderingen (voor gedetailleerdere eisen zie par. 3.3.1).</u>	
UC-VHR	Astris moet aan bedienapplicaties de mogelijkheid bieden om (meerdere) Trdl-verhinderingen voor rijweginstelling aan te brengen op seinen, secties, wissels en kruisingen.

Figuur 15 Systeemeis ASTRIS-UC-VHR uit System/Subsystem Specification'. Bron: ASTRIS System/Subsystem Specification.

TC 21 Verhinderen Bediening	SSS ASTRIS [3] H3.3.1.2	ASTRIS-TG-VHB-<volgnummer>	TG ASTRIS Verhinderen
TC 22 Toestaan Bediening	SSS ASTRIS [3] H3.3.1.3	ASTRIS-TG-TSB-<volgnummer>	TG ASTRIS Verhinderen
TC 23 Verhinderen Rijweginstelling	SSS ASTRIS [3] H3.3.1.4	ASTRIS-TG-VHR-<volgnummer>	TG ASTRIS Verhinderen
TC 24 Toestaan Rijweginstelling	SSS ASTRIS [3] H3.3.1.5	ASTRIS-TG-TSR-<volgnummer>	TG ASTRIS Verhinderen

Opdracht Verhinderen Rijweginstelling	
Testgeval identificatie	ASTRIS-TG-VHR-01-01
Testdoel	Bedienopdrachten van opdrachtgevers die niet opdrachtbevoegd zijn, worden geweigerd.
Begin situatie	Opdrachtbevoegdheden voor verschillende bedienaars zijn geconfigureerd per type opdracht.
Actie 1	Geef de opdracht 'Verhinderen Rijweginstelling' als bedienaar die daarvoor niet is bevoegd.
Verwachte reactie 1	De opdracht wordt geweigerd op basis van de afwezigheid van de opdrachtbevoegdheid van de bedienaar voor het betreffende type opdracht.
Eind situatie	Geen wijziging t.o.v. de beginsituatie.
Controle	De opdracht wordt beantwoord met één antwoord dat aangeeft dat de opdracht niet is geaccepteerd. Uit het antwoord blijkt duidelijk waarom de opdracht is afgewezen.
	Melding: Onuitvoerbaar: Niet opdrachtbevoegd

Figuur 16 Testcase 'ASTRIS-TG-VHR'. Bron: ASTRIS-STD-KT.

In figuur 17 zien we dat in de Software Requirements Specification de software eis EMC-IF-09 simpelweg is terug te vinden door middel van een zoekopdracht.

Stand [OPTIONEEL]	De stand van een bedienbaar element waarvoor de verandering geldt, verplicht als de verandering op een wissel of kruising wordt aangebracht. Mogelijke waarden: <ul style="list-style-type: none"> o links o rechts o totaal
Rol [OPTIONEEL]	De rol waarin het sein moet worden verandering, verplicht als de verandering op een sein wordt aangebracht. Mogelijke waarden: <ul style="list-style-type: none"> o begin o eind o totaal
Reden	De reden van de aan te brengen verandering. Mag alleen letters, cijfers, spaties en leestekens bevatten.
Verbijzondering [OPTIONEEL]	De verbijzondering van de reden van verandering voor rijweg. Mag alleen letters, cijfers, spaties en leestekens bevatten.
ExtraBediening [OPTIONEEL]	Geeft aan of de verandering met of zonder extra bediening is.
BedienaarType	Mogelijke waarden: <ul style="list-style-type: none"> o Centraal o Lokaal
OverdrachtNummer [OPTIONEEL]	Contractnummer om een ElementOpdracht uit te voeren welke overlap heeft met een gebied waar de bedienaar geen bedienbevoegdheid heeft.
Opdrachtgeveridentificatie	Unieke instantienaam van de bedienapplicatie. Wordt door EMC alleen voor logging gebruikt.
OpdrachtgeverType	b.v. "TRDL" of "ARI". Wordt door de EMC alleen tbv logging gebruikt.
OpdrachtId	Unieke identificatie van de opdracht.

Eis: EMC-IF-09	
Description	Een consumer dient een verandering voor rijweg op een element aan te kunnen brengen middels een veranderingRijwegElement service request.
Implementation	Intern
Rationale	

Figuur 17 'Software eis EMC-IF-09'. Bron: ASTRIS-ELEMENTCOMP-SRS.

Ook de software eis 'EMC-FUNC-004' in figuur 18 is eenvoudig terug te vinden met een zoekopdracht in de Software Requirements Specificatie ASTRIS-ELEMENTCOMP-SRS.

Eis: EMC-FUNC-004	
Description	Een ver hinderRijwegElement service request dient verwerkt te worden zoals is gespecificeerd in Tabel 3-22: Verwerken ver hinderRijwegElement Service Request.
Implementation	Intern
Rationale	De activiteiten, alternatieven en exceptie scenario's zoals beschreven vormen gezamenlijk de afhandeling van een ver hinderRijwegElement service request.

Figuur 18 'Software eis EMC-FUNC-004'. Bron: ASTRIS-ELEMENTCOMP-SRS.

Hetzelfde geldt voor de software eis 'EC-DATA-07' zoals te zien in figuur 19.

Eis: EC-DATA-07	
Description	Van seingebruik dient de informatie te worden opgeslagen zoals is beschreven in Tabel 3-4: Data SeinGebruik.
Implementation	Intern
Rationale	De informatie beschreven in de tabel is nodig voor de functionaliteit van Astris. Hoe de informatie precies wordt opgeslagen mag door de bouwer worden bepaald, normaliseren zou een andere structuur op kunnen leveren. De structuur in de tabel wordt aangehouden bij het beschrijven van de eisen.

Figuur 19 'Software eis EC-DATA-07'. Bron: ASTRIS-ELEMENTCOMP-SRS.

In figuur 20 kunnen we zien dat een tweetal software eisen uit de Software Requirements Specification ASTRIS-ELEMENTCOMP-SRS te traceren zijn naar de testbeschrijving in de Software Test Description ASTRIS-ELEMENTCOMP-STD.

366 EMC-IF-09	ASTRIS-ELEMENTCOMP-SDD	7. Tracering	ja						EMC-TG-PARC
367 EMC-IF-10	ASTRIS-ELEMENTCOMP-SDD	7. Tracering	ja						EMC-TG-PARC
368 EMC-IF-11	ASTRIS-ELEMENTCOMP-SDD	7. Tracering			ja		Zie SIG Rapportage Metrictype Duplication		
393 EMC-FUNC-003	niet alle service operaties uitgewerkt	zie document		ja					EMC-TG-PFTSB
394 EMC-FUNC-004	niet alle service operaties uitgewerkt	zie document		ja					EMC-TG-PFVHR
395 EMC-FUNC-005	niet alle service operaties uitgewerkt	zie document		ja					EMC-TG-PFTSR

Figuur 20 'Traceability component eisen naar testdocumentatie'. Bron: ASTRIS-Traceabilitymatrix.

Zoals te zien in figuur 21 is met een simpele zoekopdracht het testgeval 'EMC-TG-PARC' terug te vinden in de Software Test Description 'ASTRIS-ELEMENTCOMP-STD'.

Identificatie	EMC-TG-PARC
Testdoel	Controle parametercombinaties service requests elementservice
Testvoorschrift	<p>Equivalentieklassen:</p> <p>A. Service request: alle externe requests (IB, VHB TSB, VHR TSR, OWB, HWB)</p> <p>B. Parameters in het request</p> <p>Alle mogelijke waarden voor een parameter moeten geraakt worden (behalve uiteraard de elementidentificatie). Het gaat hier alleen om statische controles (zoals in de tabellen in par 3.6.3.2 beschreven).</p>
Schatting aantal testgevallen	Ca 60
Controle	In de response op de service requests is te zien of een service request is afgewezen vanwege een foutieve parametercombinatie.

Figuur 21 'Testgeval EMC-TG-PARC'. Bron: ASTRIS-ELEMENTCOMP-STD.

De eisen EMC-IF-09 en EC-DATA-07 zijn zogenaamde componenteisen wat ook duidelijk uit de naam blijkt. Deze twee voorbeelden zijn ook eisen die zijn opgenomen in een Software Design Description. Ook hier blijkt een goede traceability gerealiseerd. De eisen zijn zoals te zien in figuur 22 opgenomen in de Software Design Description.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	SRS	Eis	SDD	SDD	Kwalificatie					Opmerking Verificatiewijze	STD COMP			
2	Eis identificatie	vervallen		Paragraaf	D	T	A	I	S		KT Testgeval		tracing-hulpmiddel	
	EC-DATA-07		ASTRIS-ELEMENTCOMP-SDD	7.Tracering					ja	Code-inspectie Zie IDL Elementnotificaties en Elementschaduw	EMC-TG-CI		(geen garanties)	
315													3	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	SRS	Eis	SDD	SDD	Kwalificatie					Opmerking Verificatiewijze	STD COMP			
2	Eis identificatie	vervallen		Paragraaf	D	T	A	I	S		KT Testgeval		tracing-hulpmiddel	
366	EMC-IF-09		ASTRIS-ELEMENTCOMP-SDD	7.Tracering	ja						EMC-TG-PARC		(geen garanties)	3

Figuur 22 'Component requirements relatie met ontwerp'. Bron: Traceabilitymatrix project ASTRIS.

Zoals in figuur 23 te zien zijn de eisen in het de betreffende Software Design Description door middel van een eenvoudige zoekopdracht terug te vinden.

I_EC_ELEMENT_SERV

De *Element Service* zie:

- [SSDD ASTRIS] Bijlage C, Externe Services C.1
 - [SRS ELEMENT] Paragraaf 3.6.3.2
- (EMC-IF-06, EMC-IF-07, EMC-IF-08, **EMC-IF-09**, EMC-IF-10, EMC-IF-29, EMC-IF-30)

Uitgezonderd de volgende notificatie service operaties:

- *wijzigingElement* zie I_EC_WIJZIGINGELEMENT_TOPIC
- *wijzigingElementOpdracht* zie I_EC_WIJZIGINGELEMENTOPDRACHT_TOPIC

4.1.2.6 Elementen gebruik

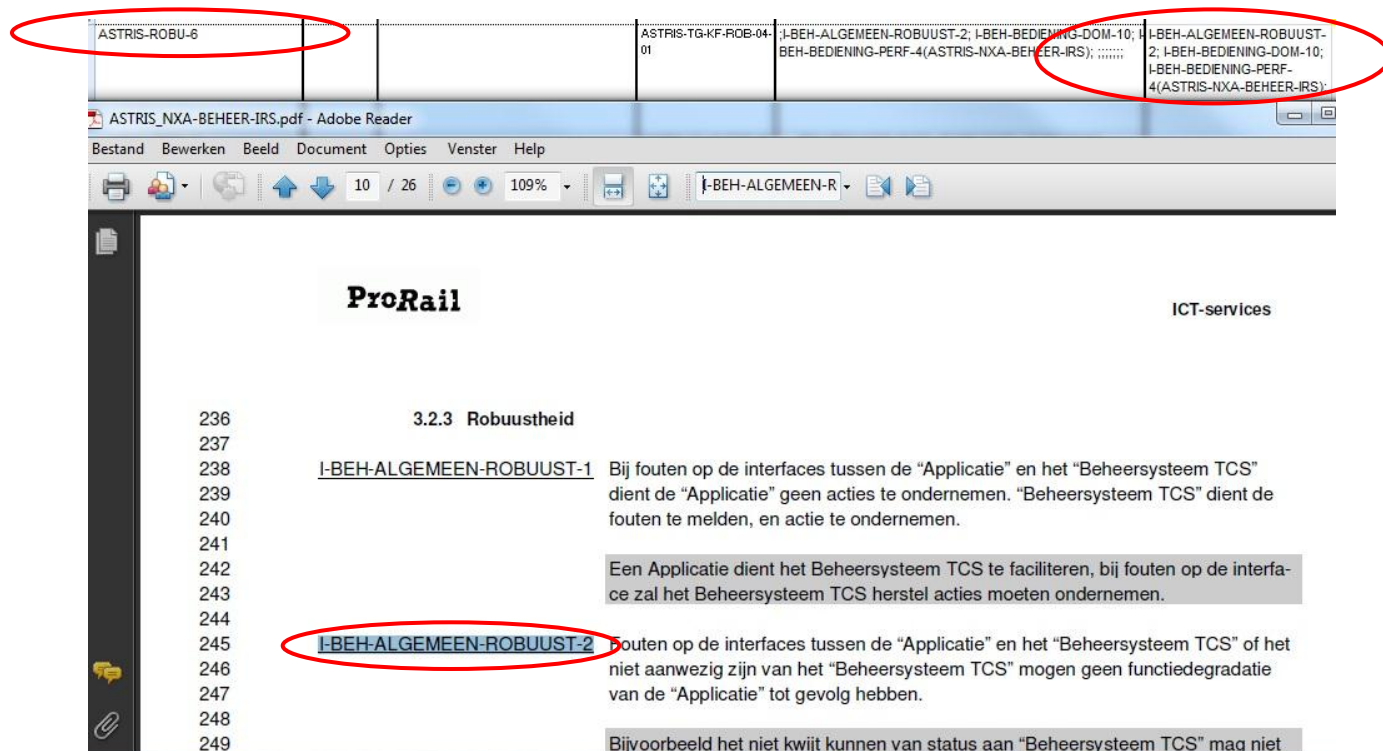
Element gebruik wordt toegevoegd of verwijderd m.b.v de Element Gebruik Service. Er zijn verschillende type gebruik, deze komen in Figuur 4-13 terug als subclasses van de *Gebruik* baseclass. Voor een volledig overzicht zie [SRS ELEMENT] paragraaf 3.5.2.2.2 en [SSDD ASTRIS] bijlage E bij "GebruiksGegevens".

Welk gebruik wordt toegevoegd is afhankelijk van de service operatie en van het Type element. Gebruik objecten van de juiste subclass worden aangemaakt m.b.v de *GebruikReserveringFactory*. Of gebruik kan worden toegevoegd, is o.a afhankelijk of het aangevraagde gebruik conflicteert met het gebruik dat reeds op gereserveerd is voor dit Element. (zie [SRS ELEMENT] EMC-FUNC-045).

De enige datamember van de *Gebruik* class is de *GebruiksGegevens* struct, deze struct is onderdeel van de IDL van het *WijzigingElementDR*. Element gebruik wordt dus toegevoegd aan het *WijzigingElementDR* (EC-DATA-06, **EC-DATA-07**, EC-DATA-08).

Figuur 23 'Component element service in SDD'. Bron: SDD project ASTRIS.

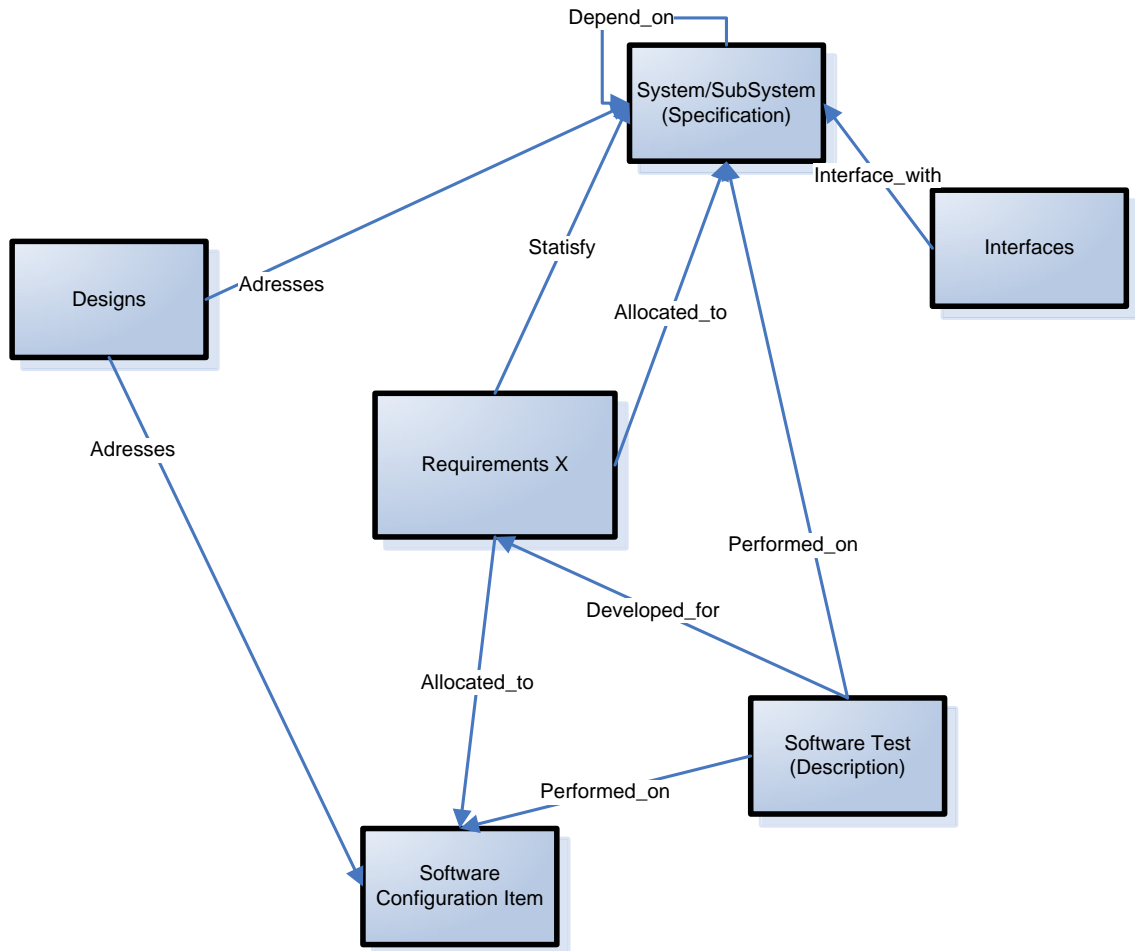
We zien dat er tracing mogelijk is tussen de Systeem eisen, software component eisen, ontwerp en de verschillende testgevallen. In figuur 24 zien we een ander voorbeeld waarbij ook de tracing naar een Interface Requirement Specification van belang is. In de eerste kolom zien we de eis zoals deze in de System/Subsystem Requirements Specification is opgenomen. In de rechterkolom zien we de Interface Requirements Specification eisen waaraan deze systeemeis is gerelateerd. Daaronder volgt dan ook één van deze eisen als voorbeeld die door middel van een zoekopdracht te vinden is in de Interface Requirements Specification.



Figuur 24 'Systeem eis ASTRIS-ROBU-6 & gerelateerde Interface eis I-BEH-ALGEMEEN-ROBUUST-2'. Bron: ASTRIS-traceabilitymatrix & ASTRIS_NXA-BEHEER-IRS

4.2 Resultaten analyse project ASTRIS

ASTRIS draait om zeer bedrijfskritische processen. Hier is gebruik gemaakt van een standaard MIL-STD-498 om de requirements te documenteren en traceability mogelijk te maken. ASTIS betreft volledig nieuwbouw en maatwerk. De Military Standard 498 is behoorlijk uitgebreid toegepast, dit kunnen we zien aan het feit dat uit elk onderdeel namelijk Planning, Specificatie, Ontwerp, Test en de Manuals en Software beschrijvingen gebruik gemaakt is van de Data Item Descriptions om de benodigde documentatie op te stellen. Dat er sprake is van een goede traceability blijkt wel uit het voorbeeld waarin een specifieke eis keurig te volgen is van specificatie tot test en vice versa. Als we dit vertalen in een conceptueel model komt deze traceability er als volgt uit te zien.



Figuur 25 'Traceability model project ASTRIS.

Het conceptueel model kent de volgende relaties:

- *Satisfy* – Requirements waarin wordt voorzien door een systeem of subsysteem.
- *Allocated_to* - Traceability van elk System requirement die toegewezen is aan die specifieke Subsystem Specification naar dat specifieke Subsystem requirement en aan Software Configuration item.
- *Depend_on* – Subsystem requirement naar het systeem welke het adresseert.
- *Developed_for* – Kwalitatieve methodes beschreven om te verifiëren of aan de requirements is voldaan. (testing/inspection)
- *Performed_on* – Traceability van elke test case in de STD naar het Systeem of het Computer Software Configuration Item requirements die het betreft. Traceability van elke Systeem of Computer Software Configuration Item welke is opgenomen in een specifieke STD.
- *Addresses* – Requirements traceability tussen de ontwerpen (database, software, interface) en de systemen en software configuration items die het systeem adresseert
- *Interface_with* – Traceability van de interface specificatie naar het betreffende System of Subsystem.

4.3 Project en domein BBMS

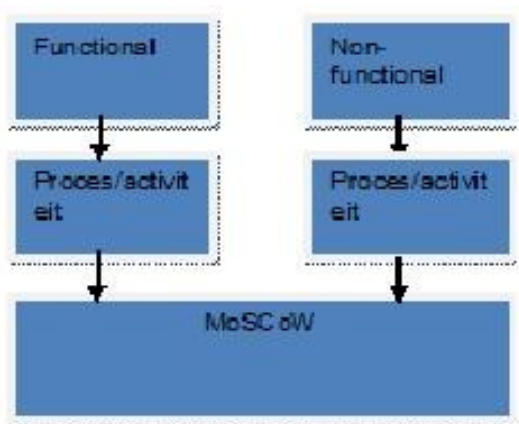
Ten behoeven van het onderhoud aan het spoor wordt meetdata en videodata verzameld over de status van het spoor. Deze data wordt gebruikt door andere partijen voor het onderhoud aan het spoor. Vanuit de wens één centraal punt van dataverzameling en beheer te hebben waarbij ProRail de regie voert in de branche is de wens ontstaan om het Branche Breed Monitoring Systeem te ontwikkelen. Het project is aanbesteed en leidt tot een leverancier die het systeem met alle bijbehorende facetten zal gaan leveren. De insteek is vooralsnog dat ProRail alles zelf zal beheren. Het systeem voorziet dus in het opslaan, en categoriseren van data en beeldmateriaal, analyseren van data en beeldmateriaal, rapporteren en het ontsluiten van data en beeldmateriaal en analyses aan derde partijen. Dit systeem zal bestaan uit een standaard pakket met daarbij de nodige uitbereidingen specifiek voor ProRail. Ook zal gewerkt gaan worden met een webeditie waarvan ProRail de eerste zal zijn die hier gebruik van maakt. Deze is dan ook grotendeels voor ProRail ontwikkeld. Het project stelt geen hoge eisen aan beschikbaarheid en beveiliging. Wel is betrouwbaarheid en integriteit van de data zeer belangrijk. Er is sprake van complexe data waarop analyses worden gedaan. Deze analyses en de resultaten dienen nauwkeurig en correct te zijn.

Omdat er sprake is van een standaard pakket en de webeditie door de leverancier ook als standaard pakket geleverd wordt is er sprake van 'slechts aanvullend maatwerk'. Dat houdt in dat op een aantal punten ProRail additionele wensen heeft waarvoor software gebouwd en/of aangepast wordt. Voor deze onderdelen zijn Functioneel Ontwerpen opgesteld. Op basis hiervan bouwt de leverancier dit maatwerk. Voorbeeld van de inhoud van een dergelijk Functioneel Ontwerp is ook opgenomen in Bijlage II.

4.3.1 Requirements structuur BBMS

In het project Branche Breed Monitoring Systeem (BBMS) is ervoor gekozen de opbouw van requirements als volgt in te delen. Men is begonnen met het onderscheid in functional en non-functional om vervolgens daarbinnen weer een opsplitsing te doen in proces/activiteit, daarna volgt de indeling naar de structuur van MoSCoW. Wat opvalt bij het project BBMS is dat de wijze van vastlegging volgens de eerder beschreven structuur geen onderscheid maakt in het soort requirement. Elke soort requirement wordt op dezelfde wijze door de beschreven boomstructuur vastgelegd.

Dit ziet er schematisch dus als volgt uit:



Figuur 26 'Requirements structuur BBMS'. Bron:

Kanttekening hierbij is wel dat wanneer je kritisch naar de requirements kijkt bijvoorbeeld niet alle requirements uit hoofdstuk 9 gedefinieerd kunnen worden als non-functionals. De structuur is dus niet heel zorgvuldig gehanteerd.

Voorbeelden hiervan zijn:

Het systeem biedt de mogelijkheid om meerdere gebruikers in een autorisatiegroep vast te leggen. De beheerder moet de autorisatie van een enkele gebruiker voor een nader te bepalen periode tijdelijk uit kunnen breiden onafhankelijk van de overige leden van de autorisatiegroep.

Hier wordt namelijk beschreven wat het systeem moet doen en niet zozeer hoe het systeem dit moet doen. Het verschil tussen functionals en non-functionals. Hetzelfde geldt voor het volgende voorbeeld.

Het systeem dient opgeleverd te worden met een online help functie. De online help functie dient in het Nederlands te zijn. Bronteksten van de online help dienen meegeleverd te worden zodat de beheerder deze kan aanpassen.

Ook zijn de requirements niet SMART. Onder SMART verstaan we in dit onderzoek 'Specifiek, Meetbaar, Acceptable, Realistisch en Tijdsgebonden'. Doel hiervan is dat de requirement helder, eenduidig en niet voor meerdere interpretaties vatbaar, geformuleerd is.

Voorbeeld van een als zodanig geformuleerde eis kan zijn:

"Het BBMS systeem biedt de mogelijkheid aan de rol 'administrator' om een onbeperkt aantal gebruikers op elk door de 'administrator' gewenst moment in een willekeurige in het BBMS systeem geconfigureerd autorisatiegroep op te nemen."

In bijlage II BBMS Requirements is te zien hoe de requirements documentatie er volgens deze structuur uitziet.

4.3.2 Requirements traceability project BBMS

In het project BBMS is geen specifieke standaard gehanteerd en geen aandacht aan requirements traceability besteed. De requirements traceability is zeer summier aanwezig doordat alle eisen genummerd zijn waarnaar verwezen wordt daar waar dit noodzakelijk werd geacht door de betrokkene. Indien er sprake was van veranderende eisen is een nieuwe versie van het pakket van eisen opgeleverd. De requirements zijn in tabellen in Word opgenomen, een vooraf bepaalde structuur van deze documenten is er niet. Wat er in dit document opgenomen moet worden is volledig naar eigen inzicht. Dit geldt ook voor de ontwerpdocumentatie zoals een Functioneel Ontwerp. De documenten zijn dan ook maar sporadisch voorzien van verwijzingen naar requirements. In het volgende voorbeeld is in een zeer beperkte mate traceability aanwezig.

Pakket van Eisen BBMS

In het pakket van eisen zijn alle eisen tekstueel beschreven, uitgewerkt in tabelvorm.

BBMS bevat meetdatasets. Meetdatasets zijn voorzien van metadata waarin informatie over het tijdstip en de locatie van de meting zijn opgenomen alsmede de wijze waarop de meting is uitgevoerd. In de eisen worden met de term "meetdata" zowel meet- als metadata bedoeld

Distributie asset configuratie (2.1)	
2.1.1	Per mail worden notificaties uitgestuurd conform een vooraf opgesteld distributie schema bij het actualiseren van de asset configuratie data.
2.1.2	Om inzicht te houden in de te verwachten meetdata en de realisatie van de gecontracteerde meetdiensten wordt op de asset data in verschillende lagen de dekking aangegeven voor de op te nemen meetdata. Deze lagen zijn: Contract Planning Uitvoering Oplevering meetgegevens Dekking aanbrengen houdt in dit verband in het per meetdataset of parameter van een meetdataset aan een laag toewijzen van delen van de railinfrastructuur. BBMS is ingericht om op de asset data in 4 lagen dekking aan te kunnen brengen
2.1.3	Daar het contract kan bestaan uit meerdere metingen per jaar, wordt op de dekkingslaag contract de frequentie aangegeven. Voor de dekkingslagen planning, uitvoering en meetgegevens zijn per laag meerdere dekkingen per jaar beschikbaar.
2.1.4	Dekkingslagen kunnen zowel handmatig aangebracht worden in BBMS zelf als geïmporteerd uit een externe bron.
2.1.5	Het handmatig vullen van een laag wordt gedaan op grafische kaarten door het aanklikken van de delen van de railinfrastructuur (per USIS-id of alle USIS-id's binnen een door de gebruiker aan te duiden bereik), waarmee dat element toegewezen wordt aan een van de dekkingslagen.
2.1.6	Dekkingslagen kunnen eenvoudig toegewezen worden aan individuele parameters, aan een meetdataset of aan meerdere meetdatasets tegelijk.

Figuur 27 'Pakket van eisen BBMS'. Bron: PvE 1.0 BBMS.

Voor een aantal onderdelen is een functioneel ontwerp opgesteld. In dit functioneel ontwerp wordt dan verwezen naar de eis in het Pakket van Eisen of wordt deze in het functioneel ontwerp opgenomen.

Functioneel ontwerp dekkingslagen BBMS

In het functioneel ontwerp is eis 2.1.3 opgenomen en wordt hiervan een uitwerking getoond. Zie figuur 28.

2.1.3	Daar het contract kan bestaan uit meerdere metingen per jaar, wordt op de dekkingslaag contract de frequentie aangegeven. Voor de dekkingslagen planning, uitvoering en meetgegevens zijn per laag meerdere dekkingen per jaar beschikbaar.
-------	---

In BBMS is het mogelijk om in een diagram het contract weer te geven, waarbij de kleur de frequentie aangeeft, zoals in het onderstaande voorbeeld is aangegeven:



Figuur 1: voorbeeld diagram weergave dekkingslagen met verschillende frequentie

Grenswaarde niveau	Voorwaarde
<input checked="" type="checkbox"/> 2x per jaar	TH = 2x per jaar
<input checked="" type="checkbox"/> 4x per jaar	TH = 4x per jaar
<input checked="" type="checkbox"/> 1x per jaar	TH = 1x per jaar
<input checked="" type="checkbox"/> Overige	TH = overige

Figuur 2: legende kleurbetekenis

Figuur 28 'Functioneel ontwerp BBMS'. Bron: FO dekkingslagen BBMS.

De traceerbaarheid beperkt zich dus tot het verwijzen naar eisen uit het Pakket van Eisen door middel van de daar toegepaste nummering. Dit gebeurt alleen in de functioneelontwerpen die opgesteld zijn.

Het tweede voorbeeld betreft een requirement aan de GIS viewer uit het pakket van eisen, zie figuur 29.

4.2.8	De GIS viewer dient over de onderstaande functionaliteiten te beschikken:
	<ul style="list-style-type: none"> - Selecteren <ul style="list-style-type: none"> o Van een of meerdere asset typen o Van een of meerdere assets - Navigeren <ul style="list-style-type: none"> o Play/Stop/Still o Forward o Reverse o GoTo, spring naar een specifiek opgegeven, bijv. kilometrering o In- en uitzoomen o Pannen - Zoeken <ul style="list-style-type: none"> o USIS ID o Op Geocode o Op Geocode + spoomummer o Op Geocode + spoomummer + kilometer (lint plus km) o Op contractgebied o Op corridor, op baanvak o Op combinaties hiervan o Op een door een gebruiker gedefinieerde indeling

Figuur 29 'BBMS requirement GIS Viewer'. Bron: PvE 1.0 BBMS.

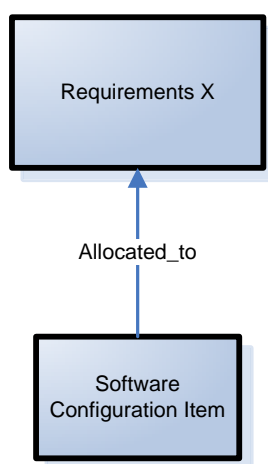
Requirement 4.2.8 is nauwelijks te herleiden tot het functioneel ontwerp dat is opgesteld. In figuur 30 zien we een onderdeel van het functioneel ontwerp waarin enigszins een onderdeel van de requirement teruggevonden kan worden. Namelijk de vereiste dat in- en uitzoomen mogelijk is.

e)	Zoom In (E) Standaard Zoom In functie. Zoom in op kaart. Middelpunt op scherm blijft centreerpunt.
f)	Zoom Uit (E) Standaard Zoom uit op kaart. Middelpunt op scherm blijft centreerpunt. Zoom Uit functie.
g)	Verschuiven (pannen) (E) Standaard Pan functie.
h)	Zoom naar gehele kaart (E) Standaard functie. Gehele kaart is gedefinieerd als de kaart van Nederland.

Figuur 30 'BBMS functioneel ontwerp GIS Viewer'. Bron: Functionele Specificatie gisviewer 1.9

4.4 Resultaten analyse project BBMS

In het project BBMS zien we dat de opzet van requirements documentatie zeer beperkt is. Er is een Pakket van Eisen wat niet veel meer omvat dan een lange tabel met alle requirements opvolgend genummerd. Daarnaast is er enige ontwerpdocumentatie aanwezig in de vorm van Functioneel Ontwerpen. Hierin wordt de nummering gehanteerd die in het Pakket van Eisen is gehanteerd. Dat brengt een zekere vorm van traceability met zich mee. Testdocumentatie is niet of nauwelijks aanwezig (er zijn enkele testdocumenten op het vlak van de technische werking) en bevatten geen enkele verwijzing of relatie met andere documenten. Kortom als we de traceability van het project BBMS vatten in een conceptueel model ziet dit er als volgt uit;



Figuur 31 'Traceability model project BBMS'.

De relatie die dit model weergeeft is als volgt te definiëren:

- *Allocated_to* – Traceability van het Functioneel Ontwerp van een specifiek Substelsysteem naar de requirement die aan dit ontwerp ten grondslag licht en vice versa.

4.5 Project en domein SBG++

De treindienstleider maakt voor het leiden van het treinverkeer gebruik van het systeem Procesleiding (PRL). Dit systeem bevat o.a. een planscherm (een tabellarische weergave van planregels) en een spoorbezettinggrafiek (SBG, de weergave van de geplande bezetting van perronsporen). Met de nodige ervaring/expertise van de treindienstleider wordt hiermee bewaakt of het plan op orde is zodat indien nodig ingegrepen kan worden.

Met het toenemen van het treinverkeer (zoals o.a. voorzien met de programma's Hoog Frequent Spoor en ETMET) of het invoeren van nieuwe be/bijsturingconcepten (zoals bij Klantgerichte logistiek) dient ook de ondersteuning voor de treindienstleider aangepast en

verbeterd te worden. Hiervoor worden inmiddels eerste stappen gezet, zoals bij de ontwikkeling van PRL34. Ook een vernieuwde SBG moet hier een bijdrage aan leveren.

Deze SBG++ moet voorzien in de volgende functionaliteit:

- een goed werkende SBG,
- *plus* de uitbreiding met meer sporen en tonen van samenhang tussen planregels
- *plus* het initiëren van mutaties in het plan vanuit de grafische weergave.

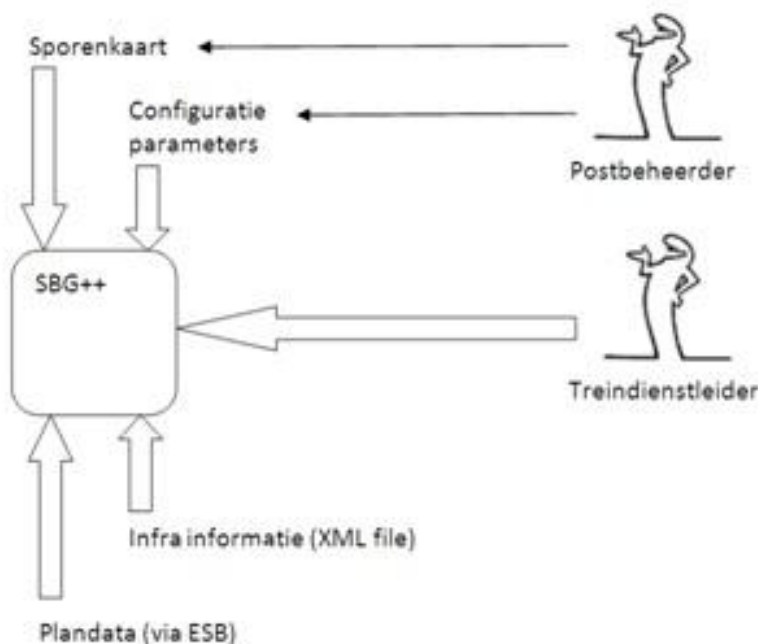
De huidige SBG in PRL is niet als basis voor een dergelijke ontwikkeling geschikt. Deze kent namelijk van meet af aan al een aantal tekortkomingen. Tevens is de mogelijkheid tot uitbreiden en herstellen van fouten in de huidige SBG ontoereikend geworden. In ieder geval is de huidige ICT infrastructuur achter PRL niet geschikt (meer) voor het ontwikkelen van de benodigde functionaliteit/grafische oplossingen. Nodig is daarom de ontwikkeling van een SBG++ met voor PRL nieuwe technologie. Behalve een functionele sprong voorwaarts brengt het SBG++ project dan ook een eerste stap in de vervanging/modernisering van de PRL omgeving door een onderdeel van PRL (de SBG) overbodig te maken. Met deze vernieuwing ontstaat weer een onderdeel/product (de SBG++) waarmee aan toekomstige wijzigingen gehoor kan worden gegeven. SBG++ 1.0 wordt ontwikkeld via de Scrum methode. Daarbij worden requirements niet vooraf opgesteld, maar tijdens de bouw. De SRS is daarom een document dat meegroeit met de gebouwde software. Wat in dit document staat is dus ook daadwerkelijk gebouwd.

De primaire rol van dit document is het onderhoud van de software. Er is gedocumenteerd welke eisen ten grondslag liggen aan de software, zodat men bij onderhoud daarop kan terugvallen. Dit document heeft nog een tweede rol: het geeft ook aan waarom een requirement nuttig is, en waarom bepaalde keuzes zijn gemaakt.

Systeemoverzicht

SBG++ vervangt het huidige SBG. SBG is het deelsysteem van Procesleiding Rijwegen dat de spoorbezettinggrafiek toont, op basis van het actuele plan. SBG++ 1.0 is read-only: het plan wordt getoond, maar de gebruiker kan het plan niet wijzigen. In latere versies van SBG++ wordt de mogelijkheid om het plan te wijzigen toegevoegd. Naast spoorbezettingen toont SBG++ ook rijwegen. Daarmee krijgt de gebruiker inzicht in de route die treinen afleggen voor en na een spoorbezetting. Waar in dit document over SBG++ wordt gesproken wordt SBG++ 1.0 bedoeld.

In het in figuur 32 getoonde context diagram worden de relaties tussen SBG++ en de buitenwereld getoond.



Figuur 32 'SBG+++ relaties buitenwereld'. Bron: SRS SBG++ 2.0 g57.

SBG++ heeft een relatie met:

- De treindienstleider.
- De postbeheerder.
- De planservice.
- Infraconfiguratiefabriek.

De treindienstleider is de primaire gebruiker van SBG++. Het geeft de treindienstleider inzicht in wanneer welk spoor door welke trein wordt bezet. Wanneer in dit document gesproken wordt over de gebruiker wordt daarmee de treindienstleider bedoeld.

De postbeheerder kan SBG++ configureren. Hij bepaalt (via de sporenkaarten) welke sporen worden getoond in SBG++. Daarnaast kan hij een (beperkt) aantal configuratieparameters instellen.

De planservice levert de informatie die SBG++ nodig heeft om spoorbezettingen en rijwegen te tonen. Alle informatie die de planservice levert is *volgens plan*. Dit kan afwijken van de uitvoering.

De infraconfiguratiefabriek levert de beschrijving (in xml vorm) van de railinfrastructuur voor zover nodig voor SBG++.

4.5.1 Requirements structuur SBG++

Het project SBG++ richt zich op het gebied van procesleiding.

Er is gebruik gemaakt van SCRUM als ontwikkelmethodiek en van de Military Standard 498. Middels SCRUM wordt gewerkt met cycles (sprints) van 3 weken. De keuzes voor deze methodes zijn organisatie (beleidsmatig) gedreven vanuit de afdeling procesleiding. Er is gebruik gemaakt van User Stories omdat dit gebruikelijk is bij het toepassen van SCRUM.

SBG++ maakt gebruik van de Military Standard 498 en enkele daardoor aangeboden Data Item Descriptions. Zo heeft men o.a. een Operational Concept Description, een Software Requirements Specificatie en een System/Subsystem Design Description opgesteld. Waar dit mogelijk werd geacht zijn documenten en inhoud gecombineerd.

Project documentatie

De documentatie wordt opgesteld tijdens de bouw. De bouw van SBG++ is een Scrum traject, waarbij User Stories worden gebruikt om de requirements te beschrijven. De Software Requirements Specificatie SBG++ 2.0 bundelt alle User Stories, en bevat ook extra informatie zoals design decisions, achtergrondinformatie enz. Daarbij wordt het oorspronkelijke User Story nummer gebruikt. Deze zijn herkenbaar aan de prefix US. In deze Software Requirements Specificatie zijn de functionele requirements beschreven in meerdere hoofdstukken. Dit is gedaan omdat deze Software Requirements Specificatie veel meer functionele requirements bevat dan overigen. Alle overige requirements staan in hoofdstuk 8, volgens de indeling die voor een SRS gebruikelijk is.

Het Software Requirements Specification SBG++ 2.0 bevat geen lege paragrafen. (Lege paragrafen zijn paragrafen waarbij expliciet wordt vermeld dat er over het betreffende onderwerp niets te melden valt.) Sommige van de genoemde features zijn nog niet daadwerkelijk gebouwd. Ze staan wel in dit document omdat ze waarschijnlijk binnenkort gebouwd gaan worden. Ze zijn voorzien van de annotatie @NOTYET, om ze makkelijk te kunnen vinden.

Om een beeld te geven van hoe het e.e.a. aan requirements documentatie er uit ziet is in bijlage III 'Requirements SBG++ een aantal voorbeelden opgenomen.

4.5.2 Requirements traceability SBG++

Het project SBG++ pretendeert de Military Standard 498 te hanteren, echter als we kritisch naar de project documentatie kijken zien we dat aan de invulling van de benodigde documentatie volgens deze standaard niet is voldaan.

Op een aantal punten is sterk afgeweken van wat de standaard voorschrijft. Eisen zijn niet concreet geïdentificeerd maar hebben vaak alleen een verwijzing naar een Use Case nummer. De

hoofdstukken die aandacht aan traceability besteden zijn niet of niet afdoende ingevuld. In de Software Requirements Specification treffen we dus requirements in de vorm van Use Cases. Als we met behulp van de identificatie die hieraan gegeven is, een zoekopdracht geven in bijvoorbeeld de Interface Requirements Specification of System/Subsystem Design Description, vinden we geen resultaat. Het traceren van de requirements blijkt dus moeilijk.

Software Requirements Specification

In de Software Requirements Specification zijn de User Stories voorzien van een identificatie op de wijze zoals te zien is in figuur 33. We zien hier User Storie US039. De prefix US wordt gehanteerd voor User Storie en daaropvolgend wordt een oplopende nummering gehanteerd. Opvallend genoeg wordt deze alleen gebruikt in een index in dat zelfde document om aan te geven waar welke User Storie terug te vinden is. Dit is niet opgesteld zoals de Military Standard in de Data Item Description van een Software Requirements Specification omschrijft.

Voorbeeld van een specificatie in de SRS:



Figuur 33 'Voorbeeld Software Requirements Specificatie wijze van User Stories Identificatie'
Bron: SRS SBG++.

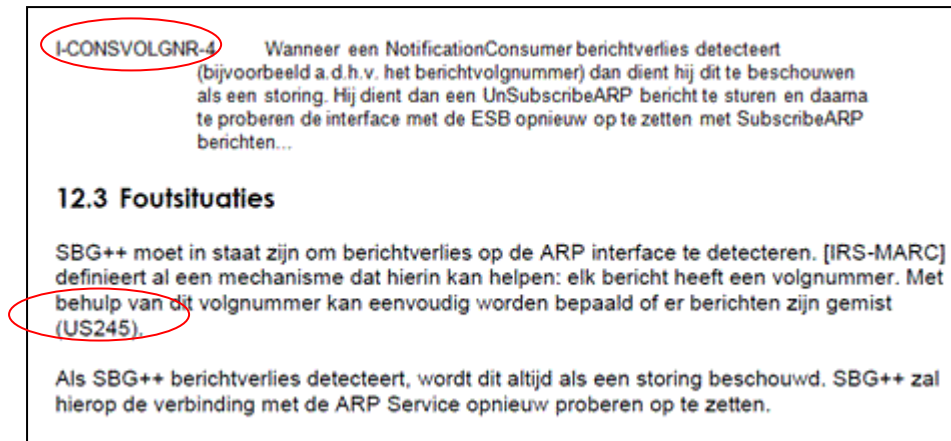
Ook valt nogal wat aan te merken op de wijze waarop de User Stories zijn gebruikt om de requirements in het SRS op te stellen. De Military Standard geeft, weliswaar kort, aan dat de requirements zo opgesteld dienen te worden dat de requirements testbaar zijn. De mate van detail bij de requirements moet er vooral opgericht zijn dat het de condities beschrijft waarop de oplevering van de requirements wordt geaccepteerd. Als we naar bovenstaand voorbeeld kijken dan heeft deze User Storie meer weg van een gebruikershandleiding dan van een concrete Requirements Specificatie.

In de andere requirement documentatie zoals de Interface Requirements Specification, Software Design Description of Interface Design Description wordt een ander soort een identificatie gehanteerd. Deze is niet gelijk aan die van de Software Requirements Specification. Er is dus geen tracering tussen de verschillende soorten specificatie en ontwerp documentatie. Daardoor is de traceability zoals deze wordt geadviseerd vanuit de Military Standard 498 niet aanwezig. Het volgende voorbeeld illustreert dit door de User Storie US039 beschrijving uit de Software Requirements Specification te traceren. Er zijn geen verwijzingen opgenomen in de specificatie en ontwerpdocumentatie naar de Software Requirements Specification, er is geen andersoortige relatie gelegd en een overkoepelende traceability matrix ontbreekt. Daarom kan niet gezocht worden doormiddel van een zoekopdracht. We hebben dan ook geprobeerd door de ander documentatie door te lezen tekstuele overeenkomsten te zoeken waarbij het aannemelijk lijkt dat er een relatie is met de requirements US039 uit de Software Requirements Specification. Dit is niet gelukt waarmee aangetoond wordt dat de traceability niet aanwezig is.

Dezelfde tracering is uitgevoerd met nog een requirement. In dit geval betreft het de requirement US245 uit de Software Requirements Specification, zie figuur 34.

Interface Requirements Specification

Omdat er geen verwijzingen opgenomen zijn in de Interface Requirements Specification naar de Software Requirements Specification of een andersoortige relatie is en een overkoepelende traceability matrix ontbreekt kan niet gezocht worden doormiddel van een zoekopdracht. Door op basis van interpretatie van beschrijvingen en hoofdstuk indelingen door het documenten te bladeren stuiten we op de requirement 'I-CONSVOLGNER-4' waarvan we vermoeden dat deze gerelateerd is aan User Storie US 245. Dit is slechts een vermoedde op basis van eigen interpretatie. Van goede traceability is dan ook geen sprake.



Figuur 34 'Voorbeeld identificatie User Stories in SRS en requirements in IRS'. Bron: SRS & IRS SBG++.

System/Subsystem Design Description

Vanuit de System/Subsystem Design Description wordt verwezen op documentniveau naar de gerelateerde interface beschrijvingen. Dit is echter alleen een verwijzing naar een document. De systeem requirements op zichzelf zijn niet voorzien van een identificatie en niet gelinkt aan de interface requirements in de Interface Requirements Specification.

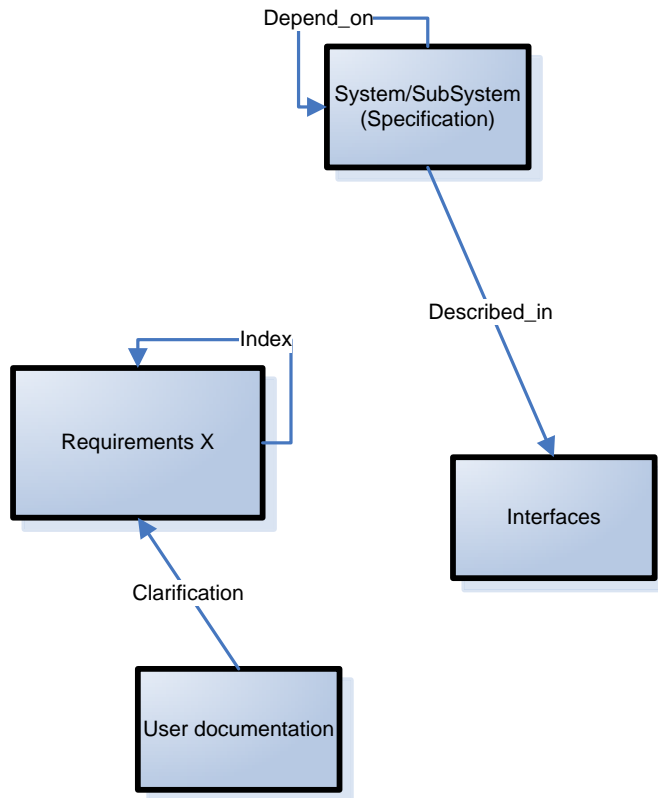
Software User Manual Description (SUM)

Opvallend genoeg is er wel sprake van traceability door middel van de nummering van User Stories naar de Software User Manual description. In de Software User Manual zijn namelijk een aantal User Stories verwerkt als uitleg. Hierover zeg de Military Standard 498 niets.

4.6 Resultaten analyse project SBG++

SBG++ is niet bedrijfskritisch maar wel zeer belangrijk voor de ondersteunende taken van Procesleiding en heeft complexe relaties met andere projecten. De functionaliteit is bij SBG++ erg belangrijk en de gebruikers betrokkenheid staat hoog in het vaandel. Het project heeft getracht de Military Standard 498 toe te passen. Echter zoals in paragraaf 4.3.5 zichtbaar is gemaakt is het traceren van de requirements tussen documentatie niet mogelijk. Als oorzaak wordt gegeven het feit dat bijvoorbeeld de System Requirements Specificatie door meerdere personen is geschreven en de opzet gemaakt zou zijn door iemand met eigen opvattingen over een System Requirements Specificatie.

Ook maakt SBG++ geen gebruik van een overkoepelende traceability matrix of een ander document dat de relaties inzichtelijk maakt. De standaard is dus slecht gevolgd want de leidraad in de Data Document Descriptions (DID's) zijn niet of nauwelijks opgevolgd. Er is dan ook geen sprake van traceability anders dan de 'eigen' gecreëerde traceability van de System Requirements Specificatie naar de Software Users Manual. Als we dit in een conceptueel model weergeven ziet de requirements traceability van het project SBG++ er als volgt uit:



Figuur 35 'Traceability model project SBG++'.

Het conceptueel model omvat de volgende relaties:

- *Depend_on* – Subsystem requirement naar het systeem welke het adresseert.
- *Described_in* – Summiere verwijzing naar het document waarin de requirements van de aan het betreffende systeem of subsysteem gerelateerde interface zijn beschreven.
- *Index* – Nummering en indexering van Use Stories.
- *Clarification* – Traceability van User documentation naar de hieraan ten grondslag liggende User Stories.

4.7 Project en domein Routelint

De cultuur van de wereld van railverkeer wordt van oudsher gekenmerkt door rationaliteit, hiërarchie en discipline. Binnen dit beeld past dat verkeersmanagement in eerste plaats op planning berust. Voor operationeel verkeersmanagement betekent dit dat de conflicten zo veel mogelijk van tevoren bij het maken van de dienstregeling worden opgelost. Het treinpersoneel krijgt de dienstregeling als het ware als “order” mee en wordt geacht deze strikt uit te voeren (met andere woorden op tijd te rijden). Indien de uitvoering van een trein te veel gaat afwijken van de planning, vindt er een herplanning plaats.

De systemen die beschikbaar zijn voor de treindienstleiders hebben tot doel de infra veilig en efficiënt toe te kennen aan de gebruikers. Deze systemen bieden hen de mogelijkheden de infra te bedienen en te borgen, zodat de gebruikers er veilig toegang tot hebben. Deze taak van de treindienstleiders moet zoveel mogelijk planmatig geschieden om een hoger en landelijk consistent kwaliteitsniveau te bereiken. Tevens moet op de bestaande railinfrastructuur meer treinverkeer met een bepaalde hoeveelheid treindienstleiders verwerkt kunnen worden. Ook moeten bediengebieden flexibel toegewezen kunnen worden om in stille tijden met minder personeel toe te kunnen. De systemen die de machinisten ter beschikking staan, hebben tot doel hen in staat te stellen veilig te rijden, de opdrachten van de beveiliging op te volgen en daarbij zo veel mogelijk binnen het plan te blijven.

Behoefte aan een “nieuw” communicatiemedium

Er is behoefte aan een “nieuw” communicatiemedium voor het ter beschikking stellen van informatie, aangezien de huidige communicatiemiddelen moeilijk uitbreidbaar zijn met nieuwe informatie. Onderbouwing daarvoor:

- Infra/ seinen/ ATB: nieuwe/ extra seinen zijn zeer kostbaar en beperkt in de hoeveelheid informatie die kan worden overgestuurd.
- GSM-R: telefonische communicatie wordt ongeschikt geacht om te voorzien in de betreffende informatiebehoefte (foutgevoelig en niet effectief).

“Nieuw” communicatiemedium past in omgeving

Het “nieuwe” communicatiemedium kan in lijn worden gebracht met gerelateerde ontwikkelingen in omgeving van machinisten:

- IAM/ TSB: Bij machinisten van NS Reizigers wordt op het scherm van de Railpocket (PDA die o.a. door machinisten van NSR gebruikt worden) momenteel ook het TSB-scherm (Tijdelijke SnelheidsBeperkingen) en het dienstkaartje getoond. De nieuwe HSM-informatie moet op termijn bij voorkeur geïntegreerd met de TSB-informatie en het dienstkaartje weergegeven *kunnen* worden. Het betreft hier alleen de mogelijkheid tot een geïntegreerde weergave, overige onderdelen dienen onafhankelijk van deze informatievoorziening te worden gerealiseerd (zoveel mogelijk ontkoppeling van processen buiten de scope van HSM).
- Bij sommige vervoerders leeft de behoefte om op het scherm van de PDA ook “eigen” informatie te kunnen tonen.
- In de toekomst wil men de HSM-informatie (net als bovengenoemde informatie) integreren in de toekomstige schermen van de stuurtafel in de cabine.

Het zal opvallen dat ook vaak de term OOGST wordt gehanteerd. OOGST staat voor Oog Op het Spoor en Trein. Routelint is een onderdeel van de verzameling systemen die de scope van OOGST vormen. Vandaar dat deze naam soms gehanteerd zal worden in documentatievoorbeelden.

4.7.1 Requirements structuur Routelint

Het project Routelint hanteert de Military Standard 498 als documentatiestandaard. Het heeft gebruik gemaakt van de Data Item Description van de standaard om invulling te geven aan de specificatie en ontwerpdocumentatie. Het project omvat bijvoorbeeld een Operational Concept Description, een System/Subsystem Specification, Software Requirement Specifications, System/Subsystem Design Description, Interface Design Descriptions.

In het OCD worden de huidige situatie, wijzigingen, operationele concept voor het nieuwe systeem en de gevolgen van het nieuwe systeem beschreven. In de System/Subsystem Specification worden de eisen op het hoogste niveau beschreven. Deze zijn voorzien van een identificatienummer dat is opgebouwd uit een categorie waar de eis onder valt zoals ‘Algemeen’ (ALG) of ‘Rijden met Routelint’ (RMR) en een opvolgend nummer. Deze identificatie wordt door alle documentatie heen gebruikt. Dit hoeft overigens niet te betekenen dat de requirement achter een identificatie altijd hetzelfde zal zijn. Om dit te traceren is een traceability matrix opgenomen in de documenten.

Om een beeld te geven van hoe het e.e.a. aan requirements documentatie er uit ziet bij het project Routelint is in bijlage IV ‘Requirements Routelint’ een aantal voorbeelden opgenomen.

4.7.2 Requirements traceability Routelint

Routelint maakt gebruik van de ontwikkel- en documentatiestandaard Military Standard 498. Dit is ook duidelijk herkenbaar in de opbouw van de documenten zoals de Software Requirements Specificatie. Het document bevat precies de opbouw en inhoud zoals deze in de Data Item Description is beschreven.

We nemen in de systeem specificatie ‘OOGST-SSS-040’ een eerste requirements om te traceren. Het betreft requirement RMR.CD.2. Als we vervolgens willen weten tot welke software requirements deze

requirement leidt openen we de Software Requirements Specification 'OOGST-DST-SRS-030-renvooi' en zoeken we op 'RMR.CD.2'. We komen dan bij de traceability tabel die in het daarvoor bestemde hoofdstuk is opgenomen, zie figuur 37. Daar zien we dat de requirement 'RMR.CD.2' gerelateerd is aan diverse software requirements en vice versa.

7.1 Vanuit deze SRS naar eisen in bovenliggende SSS

Eis in deze SRS	Eis in SSS OOG op Spoor en Trein
OOGST.DST.ALG.1	OOGST.ALG.4
OOGST.DST.ENTGEG.1	OOGST.ALG.3
OOGST.DST.ENTGEG.2	OOGST.ALG.3
OOGST.DST.RL.ALG.7	OOGST.RMR.CD.2, OOGST.RMR.CD.9, OOGST.RMR.ALG.1, OOGST.RMR.ALG.3, OOGST.RMR.ALG.5, OOGST.RMR.ALG.6, OOGST.RMR.RL.1

7.2 Van aan OOGST Diensten toegewezen eisen in SSS naar eisen in deze SRS

Eis in SSS OOG op Spoor en Trein	Eis in deze SRS
(Systeemimplementatie)	OOGST.DST.II.2
(Systeembrede ontwerpbeslissingen)	OOGST.DST.ENTGEG.7, OOGST.DST.SW.3, OOGST.DST.SW.4
OOGST.ALG.3	OOGST.DST.ENTGEG.1, OOGST.DST.ENTGEG.2, OOGST.DST.ENTGEG.3, OOGST.DST.ENTGEG.4, OOGST.DST.ENTGEG.5, OOGST.DST.ENTGEG.6, OOGST.DST.ENTGEG.7, OOGST.DST.ENTGEG.8
OOGST.RMR.CD.2	OOGST.DST.TRR.5, OOGST.DST.TRR.6, OOGST.DST.TRR.7, OOGST.DST.TRR.8, OOGST.DST.TRR.9, OOGST.DST.TRR.10, OOGST.DST.TRR.11, OOGST.DST.TRR.12, OOGST.DST.TRR.13, OOGST.DST.RL.ALG.6, OOGST.DST.RL.ALG.7, OOGST.DST.RL.ALG.8, OOGST.DST.RL.ALG.9, OOGST.DST.RL.ALG.10, OOGST.DST.RL.BRL.1, OOGST.DST.RL.BRL.11, OOGST.DST.RL.BRL.12, OOGST.DST.RL.VVB.1, OOGST.DST.RL.VVB.2, OOGST.DST.RL.VVB.3, OOGST.DST.RL.VVB.4, OOGST.DST.RL.VVB.5, OOGST.DST.RL.VVB.6, OOGST.DST.RL.VVB.7, OOGST.DST.RL.VVB.8

Figuur 36 'Traceability System Requirements & Software Requirements Routelint. Bron: OOGST-SSS-040.pdf & OOGST-DST-SRS-030-renvooi.pdf Routelint.

Vervolgens doen we eveneens een zoekopdracht in het ontwerp en komen we een vergelijkbare traceability matrix tegen. In deze matrix zien we dat deze requirement van toepassing is op de Software CI 'Routelint Gebiedgebonden', zie figuur 37 (rode omlijning).

Echter, voor de realisatie is het handig om de nadruk te leggen op de specifieke CI's waar de definities worden gerealiseerd, niet waar ze worden toegepast. Met name voor de Routelint Front End applicatie is het uitgangspunt dat de informatie getoond moet kunnen worden zonder diepgaande kennis van de onderliggende concepten.

Met het oog hierop is de toewijzing in de tabel op de laatste wijze ingevuld.

Eis	Omschrijving	Van toepassing op	Toelichting
RMR.CD.1	Treincentrisch met focus op hinder van/op andere treinen	RouteLint Gebiedgebonden	Zie SSS
RMR.CD.2	Eigen Trein hinderende Andere Trein	RouteLint Gebiedgebonden	Zie SSS
RMR.CD.3	Intakkende trein	RouteLint Gebiedgebonden	Zie SSS
RMR.CD.4	Uittakkende trein	RouteLint Gebiedgebonden	Zie SSS
RMR.CD.5	Kruisende trein	RouteLint Gebiedgebonden	Zie SSS
RMR.CD.6	Vooropgaande trein	RouteLint Gebiedgebonden	Zie SSS
RMR.CD.7	Tegemoetkomende trein	RouteLint Gebiedgebonden	Zie SSS

Figuur 37 'Requirement RMR.CD.2 gerelateerd aan Software CI'. Bron: Routelint SSDD.

Ook in de Interface Requirements Specificatie is een traceability matrix opgenomen. Dit blijkt als we hier een zoekopdracht doen naar de requirement 'RMR.CD.2'. Op basis van deze matrix weten we welke requirement in de Interface Requirement Specification relevant is, zie figuur 38. Het blijkt te gaan om alle requirements die de prefix 'OOGST.I_POST_AFNEMERS.TR.' bevatten.

5.2 Vanuit bovenliggende SSS OOG op Spoor en Trein naar eisen in deze SRS

Eis in SSS OOG op Spoor en Trein	Eis in deze SRS
OOGST.ALG.3	OOGST.I_POST_AFNEMERS.ALG.2, OOGST.I_POST_AFNEMERS.ALG.3, OOGST.I_POST_AFNEMERS.ENTGEG.*
OOGST.ALG.5	OOGST.I_POST_AFNEMERS.ALG.1, OOGST.I_POST_AFNEMERS.ALG.2, OOGST.I_POST_AFNEMERS.SES.13, OOGST.I_POST_AFNEMERS.GID.1, OOGST.I_POST_AFNEMERS.GID.2, OOGST.I_POST_AFNEMERS.GID.3, OOGST.I_POST_AFNEMERS.GID.6
OOGST.RMR.CD.1	OOGST.I_POST_AFNEMERS.TR.*
OOGST.RMR.CD.2	OOGST.I_POST_AFNEMERS.TR.*

Figuur 38 'Requirement RMR.CD.2 gerelateerd aan Interface Requirements'. Bron: IRS project Routelint

Als we naar de Software Requirements Specificatie 'RMR-HHT-SRS-030' kijken dan zien we dat er keurig een hoofdstuk geweid is aan de traceability van de requirements.

In dit hoofdstuk wordt gewerkt met een aantal tabellen waarin de specificaties worden gekoppeld aan de requirement in de System/Subsystem Specification. Onderstaand voorbeeld illustreert dit mooi. De eerste afbeelding toont de requirement in de Software Requirements Specification.

TST.6 Falen van toestandsovergangen
De gebruiker dient van een gefaalde toestandsovergang een melding te krijgen waarin de oorzaak wordt vermeld.

Toelichting:
Wanneer een aanmelding niet tot stand gebracht kan worden heeft de gebruiker terugkoppeling nodig. Een overgang kan falen doordat een aanmelding wordt geweigerd door de portal of doordat een datakanaal niet opgebouwd kan worden. Zo kan een melding bijvoorbeeld zijn: "Aanmelding mislukt. Gebruiker niet geautoriseerd".
Succesvolle toestandsovergangen vergen geen melding. De gebruiker kan immers de toestand duidelijk afleiden uit het scherm (aanmeldscherm of RouteLint) dat getoond wordt.

Figuur 39 "Requirements Specificatie Routelint". Bron: SRS project routelint.

De Software Requirements Specification 'RMR-HHT-SRS-030.pdf' bevat in hoofdstuk vijf een tabel ten behoeven van de traceerbaarheid. Hierin is precies te zien aan welke bovenliggende eis in de System/Subsystem Specification de requirement die we in afbeelding 39 zien gerelateerd is.

5 Tracering van eisen

5.1 Vanuit deze SRS naar eisen in bovenliggende SSS OOGST

Eis SRS	Omschrijving	Eis SSS
TST.1	Toestanden van RouteLint systeem	EI.GEBR.1
TST.2	Niet aangemeld	EI.GEBR.1
TST.3	Aangemeld	RMR.ALG.3
TST.4	Aanmelden gebruiker	EI.GEBR.1, RMR.ALG.1, RMR.ALG.2
TST.5	Afmelden gebruiker	EI.GEBR.1
TST.6	Falen van toestandsovergangen	EI.GEBR.1
ALG.1	Opstarten RMR applicatie door gebruiker	Systeemimplementatie
ALG.2	Verbinding met OOGST Portal	RMR.ALG.7
AANM.1	Aanmeldscherm	RMR.ALG.1
AANM.2	Gebruikersgegevens	RMR.ALG.1
AANM.3	Treinnummer Eigen Trein	RMR.ALG.1
AANM.4	Keuze weergavethema	RMR.ALG.4, RMR.MMI.4
AANM.5	Knop aanmelden	RMR.ALG.1
AANM.6	Knop annuleren	RMR.ALG.1
AANM.7	Status verbinding met telecom netwerk	Systeemimplementatie

Figuur 40 "Traceability tabel SRS Routelint" Bron: SRS project routelint.

System/Subsystem Specification

Als we de System/Subsystem Specification 'OOGST-SSS-040' er op naslaan treffen we daar de op de aangegeven identificatiecode gerelateerde eis aan.

3.3.1.1 Algemeen

In deze paragraaf worden eisen genoemd m.b.t. interactie met gebruikers die gelijk zijn voor alle gebruikers.

EI.GEBR.1

Aanmelden

Gebruikers moeten in staat worden gesteld en gedwongen zich aan te melden voordat zij gebruik kunnen maken van diensten van OOGST.

Toelichting:

Het aan de hand van [7] [8] uitwerken van beveiligingsissues in nadere eisen omtrent gebruikersnamen en wachtwoorden wordt aan onderliggende documenten overgelaten.

ICT Services, TRIS, Treinbesturing

System/Subsystem Specification (SSS) OOG op Spoor en Trein (OOGST) v.4.0

3 Eisen 52/72

Figuur 41 "Aan Software Requirements Specificatie gerelateerd eis". Bron: SSS project Routelint.

System/Subsystem Design Description

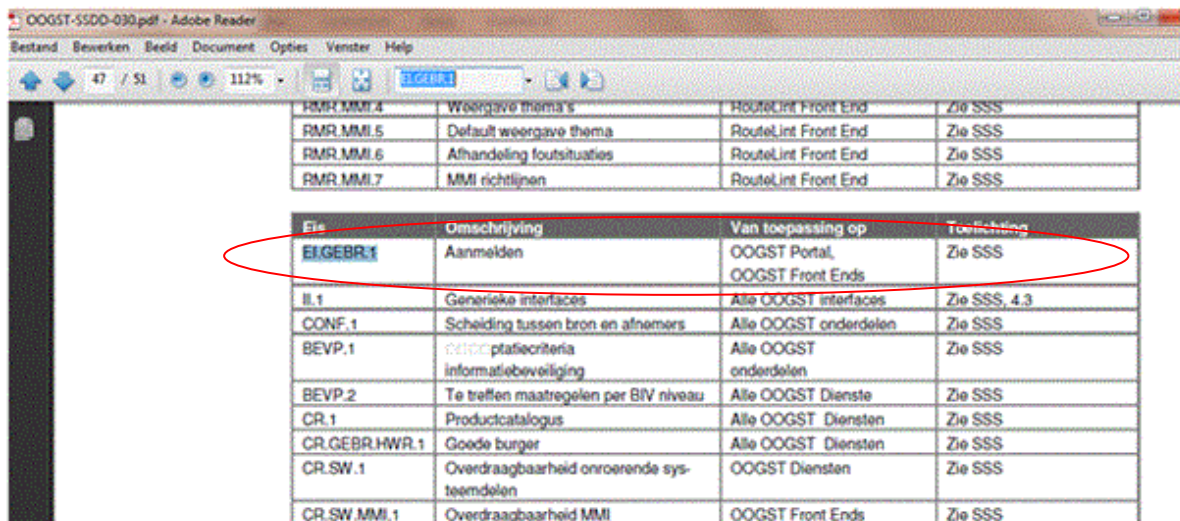
In de System/Subsystem Design Description treffen we eveneens dezelfde identificatiecode aan. Ook van specificatie naar ontwerp is requirements traceability duidelijk gewaarborgd.

4.1.2.4 Autorisatie & Autenticatie-dienst

Als onderdeel van het Linux inbeheername project is een gemeenschappelijke Autorisatie en Autenticatie-dienst ontwikkeld. Deze dienst biedt de mogelijkheid om gebruikersnaam, wachtwoorden en andere autorisaties vast te leggen. Deze functie moet ingezet worden voor de voor OOGST gebruikers noodzakelijke autorisatie en autorisatie functionaliteit. (EI.GEBR.1)

Figuur 42 "Ontwerp SSDD gerelateerd aan requirement EI.GEBR.1". Bron: SSDD project Routelint.

In de System/Subsystem Design Description wordt in het hoofdstuk voor requirements traceability de link gelegd tussen de requirement uit de System/Subsystem Specification en deze System/Subsystem Design Description (zie figuur 43).



Eis	Omschrijving	Van toepassing op	Toelichting
RMR.MMI.4	Weergave thema's	Routelint Front End	Zie SSS
RMR.MMI.5	Default weergave thema	Routelint Front End	Zie SSS
RMR.MMI.6	Afhandeling foutsituaties	Routelint Front End	Zie SSS
RMR.MMI.7	MMI richtlijnen	Routelint Front End	Zie SSS
EI.GEBR.1	Aanmelden	OOGST Portal, OOGST Front Ends	Zie SSS
IL.1	Generieke interfaces	Alle OOGST interfaces	Zie SSS, 4.3
CONF.1	Scheiding tussen bron en afnemers	Alle OOGST onderdelen	Zie SSS
BEVP.1	Acceptatiecriteria informatiebeveiliging	Alle OOGST onderdelen	Zie SSS
BEVP.2	Te treffen maatregelen per BIV niveau	Alle OOGST Dienste	Zie SSS
CR.1	Productcatalogus	Alle OOGST Diensten	Zie SSS
CR.GEBR.HWR.1	Goede burger	Alle OOGST Diensten	Zie SSS
CR.SW.1	Overdraagbaarheid onroerende systeemdelen	OOGST Diensten	Zie SSS
CR.SW.MMI.1	Overdraagbaarheid MMI	OOGST Front Ends	Zie SSS

Figuur 43 "Traceability in SSDD gerelateerd aan eis EI.GEBR.1". Bron: SSDD project Routelint.

Ook de traceerbaarheid naar testdocumentatie is in project Routelint is geborgd. Als we de Software Test Description erop naslaan vinden we keurig een traceability matrix waarin de eis 'EI.GEBR.1' gerelateerd wordt aan de testgevallen. Zie figuur 44.

In de onderstaande tabel staan de eisen uit IRS'en de interface tussen OPO en OD en tussen OD en de HHT en SSS die gesteld worden aan het systeem Rijden met Routelint. Per eis is aangegeven in welke testgevallen deze worden getest.


Eis	Testcase
OOGST.ALG.1	4.1.2
OOGST.ALG.5	4.2.3
OOGST.CR.COM.2	4.2.2
OOGST.EI.GEBR.1	4.3.1, 4.3.2
OOGST.KWAL.4	4.1.2, 4.3.3

Figuur 44 'Eis EI.GEBR.1' gerelateerd aan testgevallen. Bron: OOGST-P210-DST-OFE-STD-IT

We zien in figuur 44 de koppeling tussen de eis 'EI.GEBR.1' en de paragraaf waar het gerelateerde testgeval wordt beschreven. De afkorting OPO staat voor 'OOGST P21 Ontsluiting', OD staat voor 'OOGST diensten' en HHT 'Hand Held Terminal'.

Voorbeeld van een testgeval zien we in figuur 45. We zien daar een testbeschrijving waarin verschillende eisen relevant zijn zoals dus de 'EI.GEBR.1'.

4.3.1 LTG_IT_007 Tonen RouteLint

Testcase:	LTG_IT_007
Doel:	Aantonen dat de HHT actuele routelinten ontvangt.
Omschrijving:	Deze test laat zien dat de hele keten in staat is een simpel routelint te produceren voor 1 trein.
Testbasis:	OOGST.ALG.1, OOGST.ALG.5, OOGST.RMR.CD1, OOGST.RMR.ALG.1, OOGST.RMR.ALG.3, OOGST.RMR.ALG.5a, OOGST.RMR.ALG.6, OOGST.RMR.RL.8a, OOGST.RMR.RL.11, OOGST.EI.GEBR.1
Toestand:	PRL is actief. BITS is geconfigureerd en geïnitialiseerd. OPO is actief. OD is actief XMPP is actief. De LDAP server bevat een gebruiker test1 met als password test1. De routelint applicatie op de HHT draait niet.
Actie 1:	Plaats met BITS een trein 111 voor sein 1112 in Ddr.
Resultaat 1:	PRL toont op spoor RA een sectie bezetting en het treinnummer 111.
Actie 2	Start de Routelint applicatie op de HHT met de volgende settings: <ul style="list-style-type: none"> http server = http://kwint8.movares.nl/oogst-dst-portal id/password = test1/test1 treinnummer = 111
Resultaat 2:	De HHT toont het routelint van trein 111. 

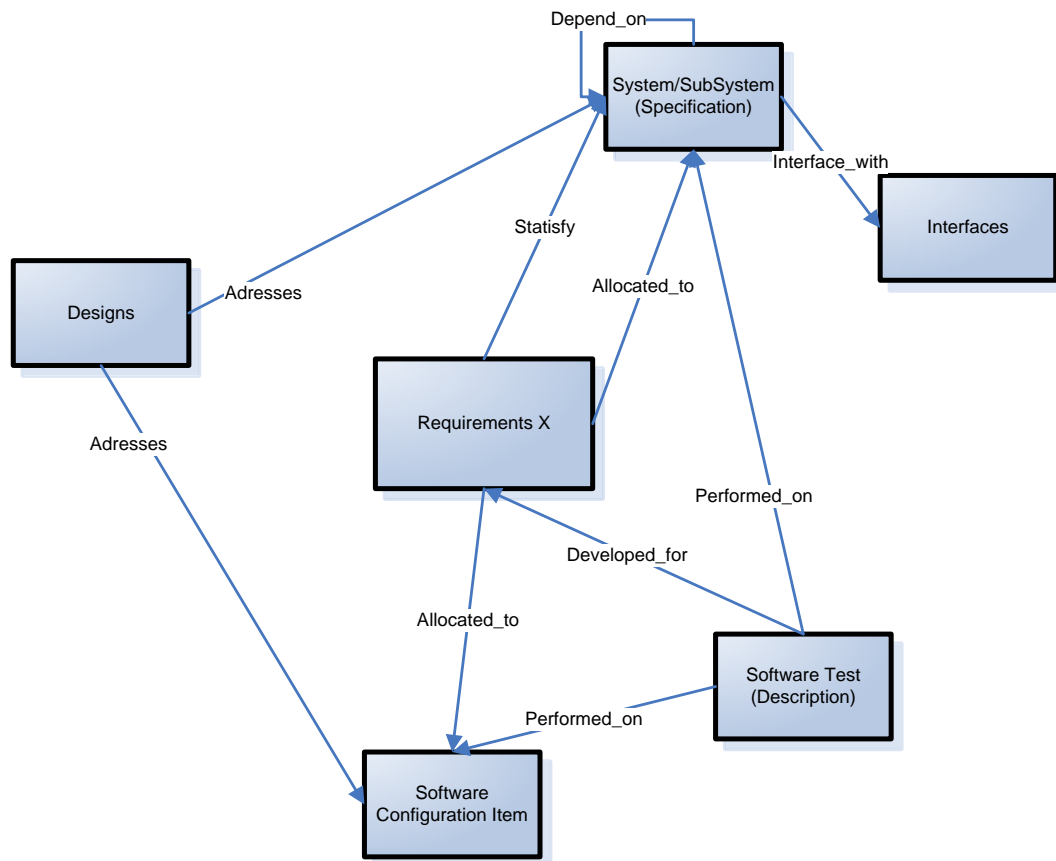
Figuur 45 'Testgeval gerelateerd aan EI.GEBR.1'. Bron: OOGST-P210-DST-OFE-STD-IT

4.8 Resultaten analyse project Routelint

Routelint heeft zeer nauwgezet de Data Item Descriptions van de Military Standard 498 gevolgd. De requirements zijn te traceren door specificaties, ontwerpen en testdocumentatie. Er is een unieke identificatie gehanteerd die overal terugkomt.

Kanttekening is wel dat gezien de hoeveelheid verschillende onderdelen een overkoepelende traceability matrix zoals we die bij project ASTRIS zagen de traceability verbeterd zou hebben.

In figuur 46 zien we de requirements traceability zoals deze in project Routelint is gerealiseerd in een conceptueel model gevat.



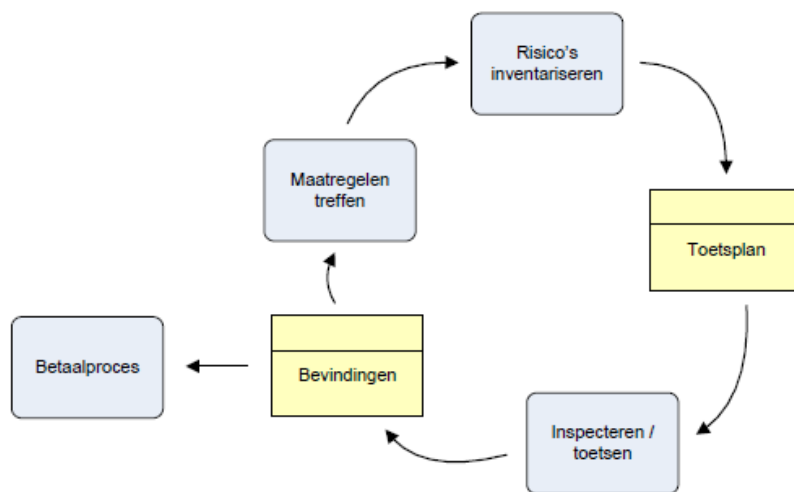
Figuur 46 'Traceability model project Routelint'.

Het conceptueel model omvat de volgende relaties:

- *Satisfy* – Requirements waarin wordt voorzien door een systeem of subsysteem.
- *Allocated_to* - Traceability van elk System requirement dat toegewezen is aan die specifieke Subsystem Specification naar dat specifieke Subsystem requirement en aan Software Configuration item.
- *Depend_on* – Subsystem requirement naar het systeem welke het adresseert.
- *Interface_with* – Requirements betreffende een specifiek Subsystem.
- *Developed_for* – Kwalitatieve methodes beschreven om te verifiëren of aan de requirements is voldaan. (testing/inspection)
- *Performed_on* – Traceability van elke test case in de STD naar het Systeem of het Computer Software Configuration Item requirements die het betreft. Traceability van elke Systeem of Computer Software Configuration Item welke is opgenomen in een specifieke STD.
- *Adresses* – Requirements traceability tussen de ontwerpen (database, software, interface) en de systemen en software configuration items die het systeem adresseert
- *Interface_with* – Traceability van de interface specificatie naar het betreffende System of Subsystem.

4.9 Project en domein RRCB

Met de introductie van het Prestatie Gericht Onderhoud of ook wel PGO-contract in 2008 is een verandering ingezet, waarbij ProRail het uitbesteedde kleinschalig onderhoud aanstuurt en betaalt op basis van de geleverde prestatie. Met het PGO-contract is tevens een uniform contractbeheersingsproces ingevoerd, waarbij de verantwoordelijkheden voor de beheersing van de risico's helder zijn belegd. De systematiek voor de contractbeheersing maakt gebruik van en ziet toe op de kwaliteitsbeheersing van de opdrachtnemer (PCA). Dit toezicht is gebaseerd op een risicoprofiel dat per contract kan verschillen. De wijze waarop de contractbeheersing door ProRail plaatsvindt is uniform, de uitvoering ervan is afhankelijk van de aanwezige risico's binnen een bepaald contract. Periodiek wordt aan de hand van de risico's een toetsplan opgesteld om voor een opvolgende periode de toetsingen vast te leggen. De rol van een tracéteam (bestaande uit een tracémanager, risicomanager (assistent tracémanager), een hoofdtoetsers, meerdere toetsers en een auditor) is het door middel van toetsingen toezien op de mate waarin de PCA zijn verplichtingen nakomt. Deze toetsingen vinden plaats op verschillende niveaus: van het kwaliteitssysteem (systeemtoetsen) via het keuringsproces (processtoetsen) tot de infra-producten (producttoetsen). De uitkomsten van de toetsingen zijn input voor het vaststellen van de rechtmatige maandelijkse prestatiebetaling. Deze stappen vormen tezamen de continue regelkring zoals aangegeven in figuur 47.



Figuur 47 "Regelkring PGO-contract". Bron: Project Start Architectuur RRCB.

In de nieuwe situatie moeten de gebruikers ondersteund worden door adequate tooling voor het totale contractmanagementproces, dus vanaf het tekenen van het contract tot en met het afgeven van de prestatieverklaring. Het PGO-contract is ten opzichte van de voorgaande contractvorm veel zakelijker ingestoken: het wordt aanbesteed en de onderhoudsaannemer heeft meer verantwoordelijkheid omdat hij wordt afgerekend op de prestatie van de infrastructuur met een vooraf vastgesteld maandbedrag. De onderhoudsaannemer moet aantonen dat hij aan de gestelde eisen voldoet. Aan de hand van in het contract vastgestelde regels kunnen inhoudingen, kortingen, bonussen en malussen opgelegd worden. Voor elk contract worden risico's geïdentificeerd met de daaruit voortvloeiende gevolgen voor ProRail, waarvoor door de PCA beheersmaatregelen getroffen dienen te worden. Voor het toetsen van de beheersing van de risico's door de PCA wordt ieder kwartaal door de hoofdtoetsers een toetsplan opgesteld. De toetsen kunnen op verschillende niveaus plaatsvinden: van het kwaliteitssysteem van de PCA, de gebruikte processen en procedures tot en met het niveau van producten / infra-objecten, en combinaties daarvan.

De geautomatiseerde systemen moeten de administratieve organisatie als volgt ondersteunen:

- Bieden van voorzieningen om gemakkelijk en snel risico's te verwerken in een toetsplan.
- Toetsresultaten registreren en deze koppelen aan het betalingsproces.
- Het vaststellen en opvolgen van afwijkingen.
- Bieden van mogelijkheden om verschillende overzichten zowel regionaal als landelijk te genereren en gegevens uit te wisselen met andere applicaties.

4.9.1 Requirements structuur RRCB

Het project RRCB is gestart als samenwerking van ProRail en Rijkswaterstaat. Omdat de Military Standard 498 een standaard is die door Rijkswaterstaat wordt gehanteerd is ook bij RRCB met deze standaard van start gegaan. Echter in een later stadium is de samenwerking beëindigd en heeft men de standaard slechts summier toegepast. Dit omdat in de beleving van de betrokkenen de standaard erg veel 'onnodig' papierwerk met zich mee bracht.

High level business requirements zijn vastgelegd in tabellen waarbij word aangegeven wat de requirement is, wie de stakeholder is, of de huidige applicatie voorziet en wat het gevolg zal zijn. De requirements zijn voorzien van een nummer. De high level business requirements zijn nader uitgewerkt tot Use Cases. Daarnaast zijn een System/Subsystem Specification en een Interface Requirements Specification opgesteld.

Om meer inzicht te geven in deze vastlegging van requirements bij het project RRCB zijn enkele voorbeelden opgenomen in bijlage V 'Requirements RRCB'.

4.9.2 Requirements traceability RRCB

In dit project blijken slechts een SSS en IRS op de Military Standard 498 gebaseerd.

Als we een tweetal requirements van het project RRCB proberen te traceren ziet dat er als volgt uit.

High level requirements

In RRCB is men van start gegaan met het vastleggen van de high level business requirements. Deze vastlegging is gedaan in tekstuele vorm in een tabel. Uit deze tabel pakken we een High Level Requirement.

High level requirements	Stakeholder	RRCB dekt functionaliteit	Gevolg
1 Vastleggen PGO contractafspraken	Controller	Minimaal; Het bedrag voor de totale looptijd moet worden ingevoerd. Deze wordt verdeeld door het systeem naar de onderliggende maanden. Specifieke contractafspraken (bv verschillende startdata van overgang contractgebieden) kunnen niet worden vastgelegd	<ul style="list-style-type: none">als de geautomatiseerde verdeling lineair niet juist is dan kan dit handmatig worden aangepast. Dit is echter niet efficiënt en is foutgevoelig.systeem is star

Figuur 48 'High level requirement RRCB'. Bron: High Level Requirements RRCB versie 13 okt.

De eerste high level requirement is het kunnen vastleggen van PGO contract afspraken. De high level requirements zijn vervolgens uitgewerkt in Use Case modellen en uitgesplitst naar meer specifieke requirements. We bekijken of deze terug te vinden is in de Use Case documenten die zijn opgesteld.

Use Case modellen

In eerste instantie kan niet op basis van een nummering of ander soort unieke identificatie de betreffende eis terug gevonden worden in de Use Case documentatie. Echter met behulp van een zoekopdracht op 'PGO' kan de eis wel teruggevonden worden. Het blijkt dat de eis is opgenomen als 'sub' eis in Use Case 1 met als identificatie 1.1. Zie figuur 48.

Actor(en)	Contractbeheerder
Doel	Vastleggen contractafspraken PGO zodat prestatieverklaring conform afspraken wordt opgesteld.
Omschrijving	Voer kenmerken van contract in, zoals duur, ingangsdatum, betaalposten, betaalposttermijnen, vaste mutanten, startdata van overgang contractgebieden.
Trigger	Getekend PGO contract
Preconditie	Opdrachtnemergegevens zijn vastgelegd (SAP)
Postconditie	Contractgegevens en afspraken zijn vastgelegd.
Requirements	<p>1.1. Het moet mogelijk zijn om PGO contractafspraken vast te leggen in het systeem</p> <p>1.2. Ieder contract heeft een contractbeheerder en een contracteigenaar (opdrachtgever), in het systeem worden contactgegevens over deze personen vastgelegd</p> <p>1.3. Een contract heeft een opdrachtnemer; de contactgegevens zijn inzichtelijk</p> <p>1.4. Een contract gaat over een onderhoudsgebied, de SAP code is inzichtelijk</p> <p>1.5. Per contract kunnen contactpersonen en hun rollen opgevoerd worden.</p> <p>1.6. Contractgegevens mogen alleen door actor Controller worden ingevoerd, gewijzigd of verwijderd worden.</p> <p>1.7. Lopende contracten kunnen niet verwijderd worden en alleen bepaalde velden kunnen nog gemuteerd worden</p> <p>1.8. Mutaties van contractgegevens moeten traceerbaar zijn en voorzien van verantwoording</p> <p>1.9. Gemaakte fouten moeten te hersteld zijn (beheerst), gelogd zijn(traceerbaar, verantw bv. teamleider of controller.</p> <p>1.10. Afspraken over maandelijkse inhouding/vergoeding worden in het systeem vastgelegd met ingangsdatum conform Art 9.1 t/m 9.7 :</p> <p>1.11. Wijzigingen op betalingsafspraken gedurende de contractperiode is mogelijk. Daarbij wordt een ingangsdatum vastgelegd zodat het achteraf herleidbaar is per wanneer en door wie vastgelegd.</p> <p>1.12. Bij een vaste mutant (uit Artikel 9.1 – 9.7) wordt een percentage opgevoerd welke invloed heeft op de betalingen richting opdrachtnemer. Dit is ter informatie; de berekening vindt buiten het systeem plaats.</p> <p>1.13. Een vaste mutant kent een tijdelijke mutant per maand; Dit is het percentage dat de betaling bepaald aan de hand van de prestaties van de opdrachtnemer. (verdere uitwerking UC 12)</p> <p>1.14. Inzicht in de wijzigingen van vaste mutanten per contract incl. ingangsdatum</p> <p>1.15. Signaal bij aanbreken Toetsmoment en betaalmoment. (nice to have: verdere uitwerking nodig)</p> <p>1.16. Na accorderen PV en na betaling mogen bedragen niet meer te muteren zijn in RRCB, aansluiting op bedragen in SAP</p> <p>1.17. Inflatiecorrectie moet in contract opgenomen kunnen worden</p>

Figuur 49 'Use Case model RRCB' Bron: UseCasemodel en Business Requirements RRCB.

System/Subsystem Specification

In het project RRCB is een System/Subsystem Specification opgesteld. Deze voldoet niet volledig aan wat de Military Standard 498 vraagt. Er ontbreken hoofdstukken waaronder die van traceability. Wel zijn keurig alle eisen in hoofdstuk 3 opgenomen zoals de Data Item Description voor een System/Subsystem Specification uit de Military Standard 498 vereist. Deze zijn opgenomen of in tabel vorm of als Use Case. De requirement die we in dit voorbeeld volgen is in de System/Subsystem Specification uitgewerkt in tabel vorm.

Ook hier geldt dat de eis niet te vinden is op basis van een unieke identificatie. De nummering die hier gehanteerd wordt wijkt namelijk weer af van die in het overzicht met High Level Business requirements en ten opzichte van de Use Case documentatie is weer een extra 'sub' niveau met nummering toegevoegd. De specificatie wordt dus wel volgens Use Case nummering weergegeven, zij het dat bij de onderliggende eisen een iets andere notatie wijze wordt gehanteerd.

3.3 Specificaties module 1: Contract & betaalschema

Use Case 1: Opvoeren PGO-contract

Van een contract worden verschillende kenmerken vastgelegd. Dit is de basis voor het maken van de prestatieverklaring.

Eisnummer	Eis	Toelichting	Kwalificatie	MeSCoW
1-001-1	De contractbeheerder CRC (Centraal Regieteam Contracting) moet in staat zijn om in het systeem één of meer contracten aan te maken zodat hiermee de prestatieverklaring conform afspraken wordt opgesteld.	Alleen de contractbeheerder CRC heeft de rechten om een contract en een PV te definiëren en wijzigen. Contractbeheerders regionaal en controllers kunnen alleen lezen.		M
1-001-2	De contractbeheerder CRC moet in staat zijn om in het systeem één of meer contracten te wijzigen zodat hiermee de prestatieverklaring conform afspraken wordt opgesteld.			M
1-001-3	De contractbeheerder CRC moet in staat zijn om in het systeem één of meer contracten voortijdig te beëindigen indien dat nodig is door het invullen van de einddatum van het contract.			M
1-001-4	Het is in het systeem niet mogelijk om een lopend contract te verwijderen.			M
1-001-5	Het is voor de contractbeheerder CRC, de contractbeheerder regionaal en de controller mogelijk om mutaties van contractgegevens in te zien en wanneer deze zijn geïnticeerd	Wie heeft de wijziging wanneer uitgevoerd (contractbeheerder CRC). Wie heeft de wijziging wanneer geïnticeerd (controller). Autorisatie op rol		S

Figuur 50 'Requirement in SSS' Bron: RRCB SSS 1.0

Als tweede voorbeeld is een andere High Level Requirement getraceerd namelijk High Level Requirement 14. Zie figuur 51.

14 Afwijking met geaccordeerde voorstel moeten binnen de oplosperiode afgerond zijn. Het systeem bewaakt dit.	Tracémanager	Nee, kan geen oplosperiode vastleggen en daarmee ook geen bewaking.	•Geen inzicht in de voortgang van afwijkingen
---	--------------	---	---

Figuur 51 'High Level Requirement 14 RRCB'. Bron: High Level Requirements RRCB.

Het blijkt echter dat in de verschillende Use Cases deze eis niet als zodanig terug te vinden is. Wellicht zal deze er wel in verwerkt zijn maar op basis van nummering of onderwerp is deze niet terug te vinden. Voorbeeld van een Use Case die betrekking heeft op de 'Afwijking' die in de eis in figuur 51 genoemd wordt is terug te zien in figuur 52. Het lijkt aannemelijk dat de in het rood omcirkelde eis waaraan gerefereerd wordt, in de Use Case de High Level Requirement veertien betreft. Echter een directe relatie is niet te leggen.

UC9. Vaststellen afwijkingen (processtap Overzicht Bevindingen/Afwijkingen)	
Actor(en)	Contractbeheerder
Doel	Bepalen welke afwijkingen opgelegd gaan worden aan de ON
Omschrijving	Beoordelen en definitief maken van afwijkingen zodat ze naar de ON verstuurd kunnen worden. Dit doet de contractbeheerder op basis van advies afwijkingen en contractafspraken
Trigger	Advies afwijkingen worden door HT ter beoordeling klaargezet voor contractbeheerder
Preconditie	Advies afwijkingen staan klaar ter beoordeling
Postconditie	Afwijkingen kunnen opgelegd worden aan de ON
Requirement	9.1. Inzicht in adviesafwijkingen per contract 9.2. Inzicht in de contractafspraken die invloed hebben op de afwijking 9.3. De contractbeheerder maakt afwijking definitief en daarna mag de afwijking niet meer gemuteerd worden. 9.4. Per definitieve afwijking worden opleggingdatum vastgelegd door contractbeheerder 9.5. Conform contractafspraken wordt een termijn vastgelegd waarop de afwijking opgelost moet zijn. 9.6. Veiligheidsafwijkingen moeten sneller afgehandeld worden en moeten prioriteit krijgen 9.7. Bij niet akkoord advies afwijking moet de contractbeheerder een onderbouwing geven. 9.8. Inzicht in afgekeurde afwijkingen in de onderbouwing. 9.9. Inzicht in openstaande afwijking van betreffende contract en ON 9.10. Advies afwijkingen moet binnen bepaalde tijd door contractbeheerder beoordeeld worden. 9.11. Als er een afwijking ON is die gelijk is aan afwijking OG dan moet één van de twee afgesloten kunnen worden. 9.12. Een afwijking kan meerdere bronnen hebben (zie req 8.2) 9.13. Bij bron wordt meldatum vastgelegd 9.14. Een afwijking heeft een datum (deadline) waarop deze hersteld moet zijn (conform Artikel 9.8)

Figuur 52 'Use Case 9 RRCB'. Bron: Use Case model en Business Requirements RRCB.

De requirements in deze voorbeelden hebben geen relatie met de interfaces die het systeem heeft. Daarom zullen deze eisen niet terug te vinden zijn in de Interface Requirement Specification. Echter na verscheidene zoekopdrachten in zowel de Use Case documentatie als in de Interface Requirements Specificatie is het niet gelukt een relatie te leggen tussen de requirements. Zoals in figuur 53 te zien is zijn de eisen in de Interface Requirements Specificatie wel voorzien van een identificatie. Deze blijkt volledige los te staan en daarom is niet na te gaan of er eisen uit de High Level Requirements en/of Use Cases gerelateerd zijn aan de Interface Specificaties.

3.4 Eisen aan de interface

Eisnummer	Eis
I-RRCB-1	SOA architectuur is gekozen voor de interface RRCB-RiskScope en RRCB-VISI. Dit betekent dat het door ProRail gekozen platform TIBCO wordt.
I-RRCB-2	Onder SOA architectuur zijn RRCB met RiskScope en VISI ontkoppeld. Alle berichten dienen via ESB getransporteerd te worden.
I-RRCB-3	Indien er een fout optreedt bij het genereren of aanroepen van een bericht, dient een foutmelding naar ESB teruggekoppeld te worden.
I-RRCB-4	Een common data model (CDM) voor de interface RRCB-RiskScope en RRCB-VISI dient ontworpen te worden.
I-RRCB-5	Gebaseerd op CDM dient ESB het datamodel van de berichten te kunnen controleren en valideren.
I-RRCB-6	Indien het valideren van het datamodel faalt, wordt het bericht direct geweigerd en een foutmelding teruggekoppeld.
I-RRCB-7	Alle interfaces dienen technische fouten automatisch te behandelen.
I-RRCB-8	Alle interfaces dienen gedetailleerde informatie over functionele fouten weer te geven.
I-RRCB-9	Een documentstructuur per contract dient aangemaakt te worden in Sharepoint voordat de interface RRCB-Sharepoint gebruikt wordt.
I-RRCB-10	De metadatarichtlijnen over documenten van DIV dient gebruikt te worden.
I-RRCB-11	De metadata van een document dient zo veel mogelijk automatisch ingevuld te worden.

Figuur 53 'Interface eisen RRCB'. Bron: IRS RRCB.

In deze analyse van de requirements documentatie in project RRCB viel wel op dat er een matrix bestond die de relatie in kaart brengt tussen de Use Cases en betrokken stakeholders. Dit draagt wel bij aan traceability naar de bron van de business requirements. Deze matrix is opgenomen in figuur 54.

Usecase	Actor	Stakeholders / sponsor
1 Opvoeren PGO contract	Contractbeheerder	Vincent Schrama, Dominik Janssen (JanJaap van der Wal)
2 Opvoeren betaalschema	Contractbeheerder	Vincent Schrama, Dominik Janssen (JanJaap van der Wal)
3 Inlezen toetswaardige risico's	Hoofdtoetser	Jan Jaap van der Wal, Toine van de Sande, Riette Lubach
4 Toetsplan maken	Hoofdtoetser	Jan Jaap van der Wal, Toine van de Sande, Riette Lubach
5 Planning uitvoering toetsplan	Hoofdtoetser	Jan Jaap van der Wal, Toine van de Sande, Riette Lubach
6 Opvragen toetsplan	Toetser	Jan Jaap van der Wal, Toine van de Sande, Riette Lubach
7 Vastleggen bevindingen	Toetser	Jan Jaap van der Wal, Toine van de Sande, Riette Lubach
8 Bepalen adviesafwijkingen	Hoofdtoetser	Jan Jaap van der Wal, Toine van de Sande, Riette Lubach
9 Vaststellen afwijkingen	Contractbeheerder	Vincent Schrama, Jannette Vosterboach
10 Aankomsten op te leveren afwijkingen	Contractbeheerder	Vincent Schrama, Jannette Vosterboach
11 Beoordelen opgeloste afwijkingen	Hoofdtoetser	Jan Jaap van der Wal, Toine van de Sande, Riette Lubach
12 Opstellen advies prestatie verklaring	Hoofdtoetser	Jan Jaap van der Wal, Toine van de Sande, Riette Lubach
13 Goedkeuren advies prestatieverklaring	Contractbeheerder	Vincent Schrama, Dominik Janssen

Vincent
 Toine
 Riette
 Jannette / Marco Kraaijenbrink (Marco vervangt Jannette tijdens haar zwangerschapsverlof)
 Jan-Jaap

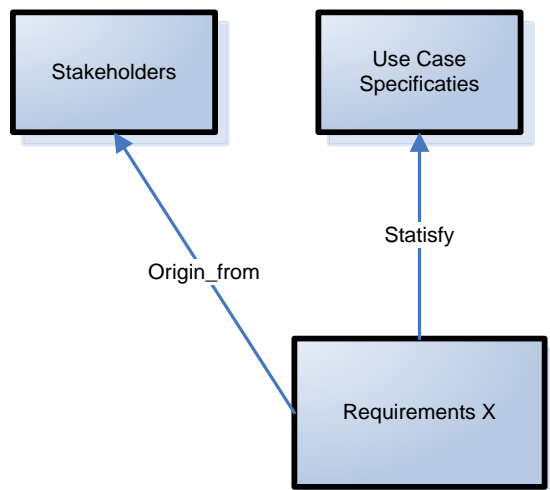
Rollen: De kleuren geven aan wie voor genoemde UC benaderd zal worden bij vragen. Diegene zal, mocht dat nodig zijn, uiteindelijke beslissingen nemen over inhoudelijke zaken. Hij/zij zal projectteamleden kunnen vragen ter ondersteuning.

Figuur 54 'Stakeholders Use Cases project RRCB'. Bron: Key users RRCB per usecase.docx

4.10 Resultaten analyse project RRCB

In project RRCB was men voornemens de Military Standard 498 te gaan volgen. Uiteindelijk is men hiervan afgeweken en is dus ook niet de Data Item Descriptions gevolgd. Alleen een SSS en IRS zijn opgesteld. Van het werken volgens de standaard is dus nauwelijks sprake. In het project is vrijwel geen traceability mogelijk. Opvallend genoeg is wel aandacht besteed aan het vastleggen van het ontstaan van de requirements door vast te leggen welke key users betrokken zijn geweest bij het opstellen van een Use Case.

Wat bij het traceren van requirements in project RRCB direct op valt is dat de nummering overal afwijkend is. Ook is de formulering en opbouw steeds afwijkend. Wel zijn de requirements genummerd en gekoppeld aan een Use Case. Er wordt geen traceability matrix toegepast. Door middel van de tabel met stakeholders uit figuur 54 is wel weer te traceren hoe een requirement is ontstaan en wie eigenaar hiervan is. Van de requirements traceability in project RRCB is eveneens een conceptueel model opgesteld. Dit model is weergegeven in figuur 55.



Figuur 55 'Conceptueel model requirements traceability RRCB'.

- *Satisfy* – Requirements waarin wordt voorzien door een specifieke functionaliteit beschreven in de vorm van een Use Case.
- *Origin_from* – Traceability van requirement naar betrokken stakeholder(s) van de specifieke requirement.

4.11 Resultaten analyse

Opvallend genoeg is er een behoorlijk verschil tussen de projecten wat betreft de structuur in het opstellen en vastleggen van requirements. Het contrast wat betreft de wijze van vastleggen van requirements bij de verschillende projecten is groot. ASTRIS hanteert een zeer gestructureerde wijze van vastleggen van requirements en hanteert een traceability matrix. Dit in tegenstelling tot het project BBMS. Zij hanteren het andere uiterste door een aantal losse tabellen met een reeks eisen op te stellen.

SBG++ lijkt hier tussen te zitten wat betreft de mate waarin requirements zijn vastgelegd. Het aantal documenten is beperkter ten opzichte van ASTRIS en bevat ook meer hoofdstukken die niet zijn ingevuld. Ook een traceability matrix ontbreekt hier. Echter ten opzichte van BBMS heeft het wel een nauwkeuriger set van documentatie en een relatie tussen specificatie en interface.

In alle projecten waarbij de Military Standard 498 toegepast is lijkt een terugkerend fenomeen, dat men niet relevante paragrafen schrapt of leeg laat. Dit terwijl de Military Standard aangeeft dat indien een paragraaf niet van toepassing is, dit expliciet als zodanig opgenomen moet worden. Je kunt hier beargumenteren dat dit met een reden gedaan is, er kan namelijk nooit onduidelijkheid ontstaan over het ontbreken van bepaalde eisen en/of opgeleverde functionaliteiten. Het effect hiervan is echter niet nader onderzocht.

Als we naar de verschillende conceptuele modellen kijken kan het volgende worden geconstateerd. De mate van en wijze waarop de Military Standard traceability toepast komt sterk overeen met het Low End Traceability model van Ramesh en Jarke (2001). Er zijn enkele nuances zoals bijvoorbeeld het feit dat de Military Standard een koppeling met de high level business requirements, welke uitgeschreven worden in een Operation Concept Description, niet kent. Dit is min of meer de relatie 'Derive' die in het model van Ramesh en Jarke (2001) terug komt.

Daarnaast besteedt de Military Standard 498 veel aandacht aan ontwerp, documentatie en de traceability tussen requirements en de ontwerpen. Dit is de relatie 'Adresses' die we in het conceptueel model van de Military Standard 498 zien. Verder houden zowel het metamodel als het Low End Traceability model van Ramesh en Jarke (2001) rekening met interfaces, systemen en subsystemen en testdocumentatie. Als we dezelfde vergelijking maken met de conceptuele modellen van de geanalyseerde projecten zien we dat de projecten die gebruik gemaakt hebben van de Military Standard 498, wat betreft requirements traceability hiermee grote overeenkomsten vertonen.

Project ASTRIS heeft de Military Standard 498 letterlijk gevolgd. Dat blijkt uit het conceptueel model wat één op één overeenkomt met die van de Military Standard. Voor project Routelint kunnen we hetzelfde vaststellen. Uitzondering in de analyse is dus SBG++. Dit project heeft ook de Military Standard 498 gevolgd maar tijdens de analyse bleek al dat niet alle documenten waren opgesteld zoals de Data Item Descriptions voorschrijven. Aan het conceptueel model is dan ook duidelijk te zien dat dit resulteert in een beperktere mate van requirements traceability.

De projecten BBMS en RRCB hebben beide weinig tot geen traceability gerealiseerd. De traceability die aanwezig is lijkt meer 'toeval' dan zorgvuldigheid. Er is geen eenduidige identificatie, relaties zijn niet gelegd en een traceability matrix is niet aanwezig. Dit is dan ook een groot verschil ten opzichte van de traceability in het Low End Traceability model van Ramesh en Jarke (2001) en van het model van de Military Standard 498. Zo ook ten opzichte van de projecten ASTRIS en Routelint. SGB++ lijkt, zoals gezegd een 'vreemde eend in de bijt'.

De verschillen die er zijn in requirements traceability blijken duidelijk uit de conceptuele modellen die zijn opgesteld. Om de verschillen tussen de modellen inzichtelijk te maken zijn twee tabellen opgesteld met daarin een vergelijking tussen de objecten die het conceptueel model bevat (zie figuur 56) en de relaties die het conceptueel model bevat (zie figuur 57).

Onderstaande tabel geeft een overzicht van de relaties die de conceptuele modellen bevatten.

Requirements Traceability literatuur model	Requirements Traceability MIL-STD-498	Requirements traceability Astris	Requirements traceability BBMS	Requirements traceability SBG++	Requirements traceability Routelint	Requirements traceability RRCB
Requirements	Requirements	Requirements	Requirements	Requirements	Requirements	Requirements
Compliance Verification Procedures	Software Test Description	Software Test Description			Software Test Description	
System_Subsystem	System_Subsystem	System_Subsystem		System_Subsystem	System_Subsystem	
External system	Interfaces	Interfaces		Interfaces	Interfaces	
	Software Configuration Item	Software Configuration Item	Software Configuration Item		Software Configuration Item	
	Designs	Designs			Designs	
				User documentation		
						Use Case Specifications
						Stakeholders

Figuur 56 'Vergelijking traceability objecten theorie, Military Standard 498 en projecten'

Onderstaande tabel geeft een overzicht van de relaties die de conceptuele modellen onderkennen.

Requirements Traceability literatuur model	Requirements Traceability MIL-STD-498	Requirements traceability Astris	Requirements traceability BBMS	Requirements traceability SBG++	Requirements traceability Routelint	Requirements traceability RRCB
Derive						
Developed_for	Developed_for	Developed_for			Developed_for	
Satisfy	Satisfy	Satisfy			Satisfy	Satisfy
Allocated_to	Allocated_to	Allocated_to	Allocated_to		Allocated_to	
Performed_on	Performed_on	Performed_on			Performed_on	
Depend_on	Depend_on	Depend_on		Depend_on	Depend_on	
Interface_with	Interface_with	Interface_with			Interface_with	
-	Adresses	Adresses			Adresses	
				Described_in		
				Index		
				Clarification		
						Origin_from

Figuur 57 'Vergelijking traceability relaties theorie, Military Standard 498 en projecten'.

Verklaring relaties:

Ramesh en Jarke (2001):

- 'Derive' – requirements verkregen uit requirements op een hoger niveau
- 'Developed_for' – requirements waarvoor de compliance verification procedures zijn ontwikkeld.
- 'Satisfy' – link tussen de requirements en het systeem component dat in deze requirements voorziet.
- 'Allocated_to' – alle requirements gerelateerd aan systeem componenten.
- 'Performed_on' – relatie tussen de 'compliance verification procedures (CVP's) en de componenten, op basis waarvan geverifieerd wordt of het component voldoet aan de requirements. Feitelijk dus tests of simulaties.
- 'Depend_on' – relatie tussen systeem componenten die afhankelijk van elkaar zijn.
- 'Interface_with' – relatie om te verifiëren hoe een requirement ten aanzien van de interface met een extern systeem wordt ingevuld.

Military Standard 498:

- *Satisfy* – Requirements waarin wordt voorzien door een systeem of subsysteem.
- *Allocated_to* - Traceability van elk System requirement dat toegewezen is aan die specifieke Subsystem Specification naar dat specifieke Subsystem requirement en aan Software Configuration item.
- *Depend_on* – Subsystem requirement naar het systeem welke het adresseert.
- *Developed_for* – Kwalitatieve methodes beschreven om te verifiëren of aan de requirements is voldaan. (testing/inspection)
- *Performed_on* – Traceability van elke test case in de STD naar het Systeem of het Computer Software Configuration Item requirements die het betreft. Traceability van elke Systeem of Computer Software Configuration Item welke is opgenomen in een specifieke STD.
- *Addresses* – Requirements traceability tussen de ontwerpen (database, software, interface) en de systemen en software configuration items die het systeem adresseert
- *Interface_with* – Traceability van de interface specificatie naar het betreffende System of Subsystem.

Project ASTRIS:

- *Satisfy* – Requirements waarin wordt voorzien door een systeem of subsysteem.
- *Allocated_to* - Traceability van elk System requirement die toegewezen is aan die specifieke Subsystem Specification naar dat specifieke Subsystem requirement en aan Software Configuration item.
- *Depend_on* – Subsystem requirement naar het systeem welke het adresseert.
- *Developed_for* – Kwalitatieve methodes beschreven om te verifiëren of aan de requirements is voldaan. (testing/inspection)
- *Performed_on* – Traceability van elke test case in de STD naar het Systeem of het Computer Software Configuration Item requirements die het betreft. Traceability van elke Systeem of Computer Software Configuration Item welke is opgenomen in een specifieke STD.
- *Addresses* – Requirements traceability tussen de ontwerpen (database, software, interface) en de systemen en software configuration items die het systeem adresseert
- *Interface_with* – Traceability van de interface specificatie naar het betreffende System of Subsystem.

Project BBMS:

- *Allocated_to* – Traceability van het Functioneel Ontwerp van een specifiek Subsysteem naar de requirement die aan dit ontwerp ten grondslag ligt en vice versa.

Project SBG++:

- *Depend_on* – Subsystem requirement naar het systeem welke het adresseert.
- *Described_in* – Summiere verwijzing naar het document waarin de requirements van de aan het betreffende systeem of subsysteem gerelateerde interface zijn beschreven.
- *Index* – Nummering en indexering van Use Stories.
- *Clarification* – Traceability van User documentation naar de hieraan ten grondslag liggende User Stories.

Project Routelint:

- *Satisfy* – Requirements waarin wordt voorzien door een systeem of subsysteem.
- *Allocated_to* - Traceability van elk System requirement die toegewezen is aan die specifieke Subsystem Specification naar dat specifieke Subsystem requirement en aan Software Configuration item.
- *Depend_on* – Subsystem requirement naar het systeem welke het adresseert.
- *Interface_with* – Requirements betreffende een specifiek Subsystem.
- *Developed_for* – Kwalitatieve methodes beschreven om te verifiëren of aan de requirements is voldaan. (testing/inspection)

- *Performed_on* – Traceability van elke testcase in de STD naar het Systeem of het Computer Software Configuration Item requirements die het betreft. Traceability van elke Systeem of Computer Software Configuration Item welke is opgenomen in een specifieke STD.
- *Adresses* – Requirements traceability tussen de ontwerpen (database, software, interface) en de systemen en software configuration items die het systeem adresseert
- *Interface_with* – Traceability van de interface specificatie naar het betreffende System of Subsystem.

Project RRCB:

- *Satisfy* – Requirements waarin wordt voorzien door een specifieke functionaliteit beschreven in de vorm van een Use Case.
- *Origin_from* – Traceability van requirement naar betrokken stakeholder(s) van de specifieke requirement.

Op basis van de analyse waarvan de resultaten in de figuren 51 en 52 vergeleken zijn blijkt dat er nogal wat verschillen in de gerealiseerde requirements traceability in projecten is. Wat is nu de verklaring voor deze grote verschillen?

Uit navraag bij betrokkenen en op basis van analyse van de projectdocumentatie blijken in ieder geval de volgende zaken ten grondslag te liggen aan de uiteenlopende structuren van requirements in de geanalyseerde projecten.

- Te hanteren/bekende standaard in domein Treindienst en ReisInformatie Systemen (TRIS) vs geen standaarden in domein Business Informatie Systemen (BIS).
- Project BBMS lijkt geen bewuste of weloverwogen keuze gemaakt te hebben om de requirements uit te schrijven. Hierbij is de keuze van de projectleider bepalend geweest. Ook is van tevoren niet nagedacht over documentatie standaarden of andersoortige conventies.
- SBG++ heeft de keuze gemaakt voor de Military Standard 498 omdat dit organisatie onderdeel gewend was om via deze documentatie standaard te werken. Daarnaast is gekozen voor het gebruik van User Stories omdat met SCRUM gewerkt wordt. De oorzaak van het niet goed opvolgen van de instructies uit de Data Item Descriptions en de aansluiting van identificaties op elkaar, ligt in het feit dat de documenten door veel verschillende betrokkenen zijn onderhouden en hierop geen sturing heeft plaatsgevonden.
- ASTRIS heeft eveneens keuze gemaakt om met de MIL-STD standaard te werken omdat hier daarmee ervaring was en men dit als 'bedrijfsstandaard' ervaart in het betrokken organisatieonderdeel.

5 Conclusie en aanbevelingen

In hoofdstuk 4 is een analyse uitgevoerd en zijn de resultaten besproken. In dit hoofdstuk worden naar aanleiding hiervan conclusies en aanbevelingen gedaan.

De onderzoeksvraag van dit project luidde:

Wat zijn de verschillen tussen de conceptuele modellen van requirements traceability in theorie, in de Military Standard 498 en in de praktijk en hoe kan traceability het best ingericht worden?

De literatuurstudie heeft als resultaat antwoordt gegeven op de vraag 'Wat wordt er verstaan onder requirements management?' en 'Wat wordt er verstaan onder requirements traceability?'. Het antwoord betreft een tweetal definities die in dit onderzoek zijn overgenomen uit de literatuur. Onder requirements management verstaan we *'The goal of requirements management is to capture, store, disseminate, and manage information. Requirements management includes all activities concerned with change & version control, requirements tracing, and requirements status tracking'*. Onder requirements traceability verstaan we *"refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases)."* Gotel en Finkelstein (1994). Ook is uit de literatuur een conceptueel model overgenomen wat een weergave is van de minimale aspecten die nodig zijn om requirements traceability in te richten. Dit is het low-end traceability model van Ramesh en Jarke (2001)

Daarnaast is de Military Standard 498 bestudeerd. De standaard realiseert de traceability doordat het concreet vermeldt, met name in de Data Item Descriptions, wat er beschreven moet worden. Het besteedt daarbij expliciet aandacht aan traceability. Ook benoemt de standaard de verantwoordelijkheid voor de ontwikkelaar ten aanzien van traceability. Op basis van deze maatregelen om requirements traceability te realiseren is tevens een conceptueel model opgesteld.

Op basis van de conceptuele modellen en het analyseren van de projecten in de praktijk kan worden vastgesteld dat een ontwikkel- en documentatiestandaard als de Military Standard 498 een mate van requirements traceability hanteert die vergelijkbaar is met die uit de theorie van Ramesh en Jarke (2001). De tabellen in figuur 56 en 57 laten dat ook duidelijk zien. Ook laten deze tabellen duidelijk zien dat wanneer de Military Standard 498 niet volledig en correct wordt gevolgd ook niet de requirements traceability ontstaat die deze standaard tracht te realiseren. Dit is op zichzelf een logisch gevolg wat vermoedelijk bij elke standaard zal op treden. In de projecten SBG++ en RRCB pretendeert men met de Military Standard 498 te werken maar zijn documenten die wel toegevoegde waarde kunnen bieden niet gemaakt en/of hoofdstukken niet gevuld. Dit is omdat de standaard vroegtijdig is losgelaten bij RRCB en er geen sturing op het gebruik van de standaard is geweest bij SBG++. Daarmee moet een kritische kanttekening geplaatst worden bij de vraag of je dus kunt spreken van projecten die de Military Standard 498 hebben toegepast of van projecten waarbij geen standaard is toegepast. Het vermoeden is, dat dit in ieder geval geen verandering teweeg heeft gebracht in de resultaten van dit onderzoek.

Daar waar de standaard wel wordt toegepast en de handvatten voor traceability worden ingevuld leidt dit zeker tot verbetering van traceability in het project. Van de projecten in de praktijk die geanalyseerd zijn, is in ieder geval duidelijk zichtbaar dat daar waar de Military Standard 498 is gehanteerd, de traceability aanwezig is. Zelfs bij de projecten waarbij de standaard slechts 'in naam' en niet volledig in uitvoering is toegepast, is een zekere mate van traceability aanwezig. Om requirements traceability te realiseren helpen handvatten die de standaard biedt, zoals een unieke identificatie van requirements en het opstellen van een traceability hoofdstuk of matrix. Dit leidt echter wel tot de vraag of het noodzakelijk is om een standaard te hanteren om invulling aan deze handvatten te geven, of kan men ook naar eigen inzicht dergelijke middelen aanwenden?

Dit onderzoek toont in ieder geval aan dat voor het goed inrichten van requirements traceability het onderkennen van elementen als requirements, systemen- en subsystemen, externe systemen, test- en verificatieprocedures, van essentieel belang zijn. Daarbij is het vooral van belang dat ook de relatie

tussen deze elementen onderkend word en de requirements in al deze elementen terug keren. Pas als dit onderkend is en er bij het managen van de requirements hiermee rekening wordt gehouden, kan traceability gerealiseerd worden.

5.1 Vervolg onderzoek.

Het bestuderen van de Military Standard 498 en requirements traceability op zowel projecten die geen gebruik hebben gemaakt van een standaard, als projecten die van de Military Standard 498 gebruik gemaakt hebben, heeft laat zien dat de Military Standard 498 een zekere mate van requirements traceability realiseert. Dit komt ook tot uiting bij projecten die de standaard volgen.

De scope van dit onderzoek heeft zich gericht op het inzichtelijk maken van de requirements traceability in theorie, de Military Standard 498 en de praktijk en de verschillen daarin. Uit het onderzoek lijkt een duidelijk verband naar voren te komen tussen een bepaalde mate van requirements traceability en het toepassen van de Military Standard 498. Interessant vervolg onderzoek kan zijn, dit in een breder scala van projecten nogmaals te toetsen, dus een meer kwantitatief onderzoek hiernaar. Zeer interessant kan ook zijn om te onderzoeken wat het effect hiervan is op het succes van een project.

5.2 Aanbevelingen

Het is belangrijk dat onderkend wordt dat bij requirements management en in het bijzonder requirements traceability belangrijk is te realiseren met welke elementen je te maken hebt. Dan wordt duidelijk dat het uniek identificeren van requirements, vastleggen van relaties en het gebruik van bijvoorbeeld een requirements traceability matrix noodzakelijk is.

Het is te adviseren een bedrijfsbrede standaard zoals de Military Standard 498 te hanteren voor requirements documentatie. De standaard draagt duidelijk bij aan de traceerbaarheid van requirements. Vooral wanneer gewerkt wordt volgens de Data Item Descriptions is traceability goed geborgd. Wel dient zorg en aandacht besteed te worden aan de wijze waarop de standaard wordt toegepast. Bij het gebruik van een standaard als de Military Standard 498 blijkt duidelijk dat er sprake is van richtlijnen die, wanneer niet gevolgd, ook weinig tot geen meerwaarde creëren. Er zal dus goed gekeken moeten worden hoe de standaard desgewenst 'op maat' toegepast kan worden zonder dat belangrijke aspecten als traceability in het geding komen.

Om dit te bewerkstelligen lijkt opleiding in het toepassen van deze standaard en vooral het 'op maat' toepassen een voor de handliggende oplossing. Daarnaast is een belangrijke rol weggelegd voor het management om de standaard op te nemen in het beleid en hierop te sturen.

Bibliografie

- Abelia Military Standard 498 Overview and Tailoring Guidebook <http://www.abelia.com>
- Cenelec (2011) European Commission for Electrotechnical Standardization, website, http://www.cenelec.eu/dyn/www/f?p=104:110:929889347772010:::FSP_PROJECT,FSP_LAN_G_ID:20508,25
- Davis, A. (2004). *Just Enough Requirements Management, Part I*. Information Systems Department, The University of Colorado at Colorado Springs Colorado Springs, Colorado 80933-7150. <http://conferences.embarcadero.com/article/32301>
- Eberlein, A en J.C. Sampaio do Prado Leite (2002). "Agile Requirements Definition: A View from Requirements Engineering". Proceedings of the International Workshop on Time-Constrained Requirements Engineering (TCRE'02) : p4-8
- Gotel, O.C.Z en C. W. Finkelstein (1994) "An Analysis of the Requirements Traceability Problem" Proceedings of the First International Conference on Topic(s): Computing & Processing (Hardware/Software) ; Signal Processing & Analysis, pp: 94 – 101.
- Gotel, O. C. Z. (1995). "*Contribution structures for requirements traceability*" (Doctoral dissertation, Imperial College).
- Gotel, O., & Finkelstein, A. (1995, March). Contribution structures [Requirements artifacts]. In Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on (pp. 100-107). IEEE.
- IEEE Std 610.121990 (Revision and redesignation of IEEE Std 792-1983), "IEEE Standard Glossary of Software Engineering Terminology". ISBN 1-55937-067-X.
- IEEE Std 830-1984 (Revision of IEEE Std 830-1993), "IEEE Recommended Practice for Software Requirements Specifications". ISBN 1-55937-395-4.
- IEEE Std 830-1998 (Revision of IEEE Std 830-1993), "IEEE Recommended Practice for Software Requirements Specifications". ISBN 0-7381-0332-2.
- ISO/IEC12207 IEEE Std 12207-2008 Second edition (2008), "Systems and software engineering - Software life cycle processes". ISBN 0-7381-5664-7
- Letelier, P. (2002, September). A framework for requirements traceability in UML-based projects. In *Proc. of 1st International Workshop on Traceability in Emerging Forms of Software Engineering* (pp. 173-183).
- Loconsole, A. (2001, April). Measuring the requirements management key process area. In *Proceedings of ESCOM-European Software Control and Metrics Conference, London, UK* (pp. 67-76).
- Lormans, M., van Dijk, H., van Deursen, A., & Nöcker, E. (2005). Omgaan met veranderende requirements in outsourcing-projecten. *Informatie (NGI)*.
- Lormans, M., Gross, H. G., van Deursen, A., & van Solingen, R. (2006, October). Monitoring requirements coverage using reconstructed views: An industrial case study. In *Reverse Engineering, 2006. WCRE'06. 13th Working Conference on* (pp. 275-284). IEEE.
- Military Standard Software Development and Documentation 498 (1994) Department of Defense United States of America http://www.everyspec.com/MIL-STD/MIL-STD-0300-0499/MIL-STD-498_25500/
- Norma A. Stopyra, (1996) "MIL-STD-498 PDF Roadmap Version 2.0 based on the MIL-STD-498". <http://www.abelia.com/498pdf/roadmap.pdf>

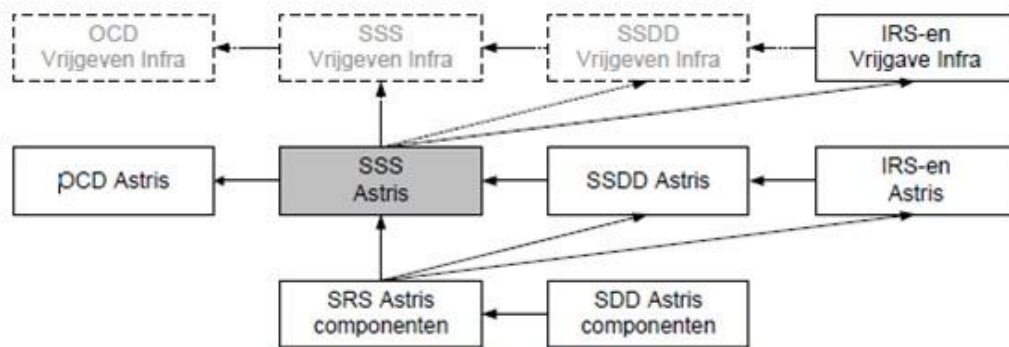
- Pandey, D., Suman, U., & Ramani, A. K. (2010, October). An effective requirement engineering process model for software development and requirements management. In *Advances in Recent Technologies in Communication and Computing (ARTCom), 2010 International Conference on* (pp. 287-291). IEEE.
- Parviainen, Päivi; Hulkko, Hanna; Kääriäinen, Jukka; Takalo, Juha; Tihinen, Maarit 2003. VTT Electronics, Espoo. <http://www.vtt.fi/inf/pdf/publications/2003/P508.pdf> 106 p.
- Peatsch F., Eberlein, A., Maurer F., (2003), Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'03) IEEE (P 308 – 313)
- Pinheiro, F. A. (2004). Requirements traceability. *KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE*, 91-114.
- Ramesh, B en M. Jarke (2001) "Toward Reference Models for Requirements Traceability". IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 27, NO. 1, JANUARY 2001, pp: 58 – 93
- Soares, M. D. S., & Vrancken, J. (2008). Model-driven user requirements specification using SysML. *Journal of Software*, 3(6), 57-68.
- Sorensen, R. (1999). Software standards: Their evolution and current state. *CrossTalk: The Journal of Defense Software Engineering*, 12(12), 21-25.

Bijlage I ASTRIS Requirements

Het project ASTRIS heeft vrij nauwkeurig de Data Item Descriptions van de Military Standard 498 gevolgd. Dit heeft geresulteerd in de volgende requirements documentatie.

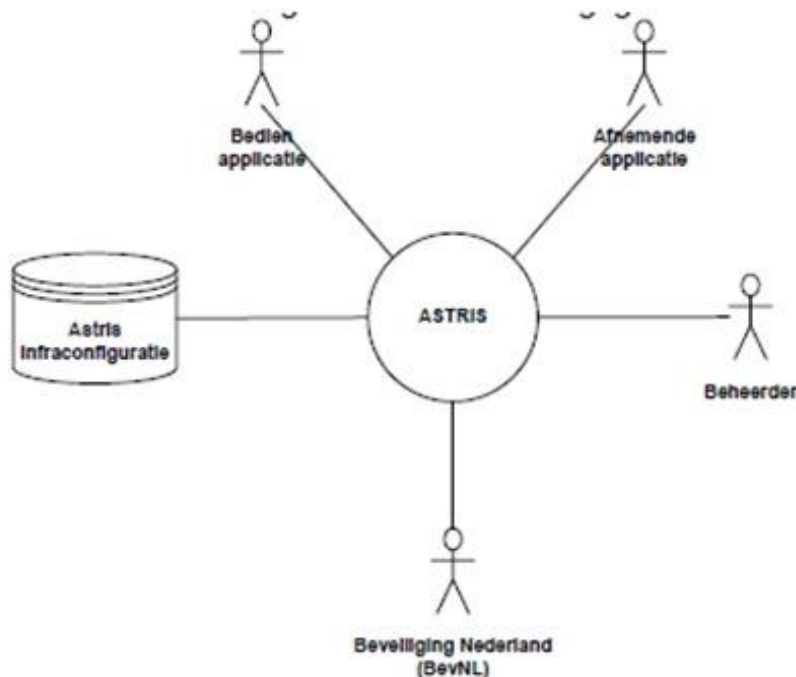
System/Subsystem Specification – ASTRIS

De Subsystem Specification (SSS) legt de eisen vast voor het subsysteem geïdentificeerd als ASTRIS en geeft aan hoe de eisen behoren te worden getoetst. Het document geeft aan waar het in de productboom te plaatsen is en geeft een overzicht van het systeem.



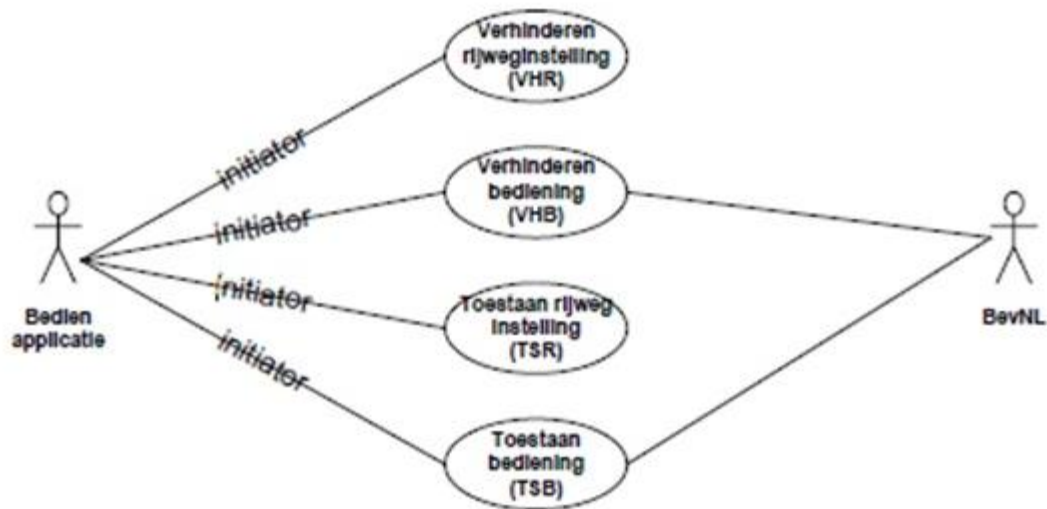
Figuur 1. Plaats van dit document in het ontwikkeltraject

Figuur 58 'Productboom ASTRIS'. Bron: SSS project ASTRIS.



Figuur 59 'Systeem context'. Bron: ASTRIS-SSS.

Het document geeft daarbij ook een beschrijving van de context van het systeem. Zoals aangegeven legt het de eisen voor het subsysteem vast. Alle functionele eisen zijn opgesteld op basis van Use Cases. Er zijn enkele functionele thema's benoemd zoals bijvoorbeeld 'Treindienstleider-verhinderingen'. Per thema is een Use Case gemaakt. Vervolgens zijn de use cases uitgewerkt in tekstvorm welke zijn omgezet in eisen. Voor 'Treindienstleider-verhinderingen' ziet dit er dan als volgt uit.



Figuur 60 'Use case thema'. Bron: ASTRIS-SSS.

Use case VHR: Verhinderen rijweginstelling

Inleiding

Deze use case beschrijft het scenario waarin de treindienstleider via de bedienapplicatie een Trdl-verhinding voor rijweginstelling aanbrengt op een infraelement. Nadat de verhinding is aangebracht zullen rijweginstelopdrachten over het betreffende element in de opgegeven stand/rol niet meer geaccepteerd worden.

Actors

Bedienapplicatie (initiator).

Normale scenario

1. De opdracht voor verhinding wordt door Astris ontvangen van de bedienapplicatie.
2. Astris controleert of de verhinding kan/mag worden aangebracht. Als dat niet het geval is wordt de opdracht afgewezen (einde use case).
3. De verhinding wordt opgeslagen in Astris
4. Astris stuurt het opdrachtresultaat (geaccepteerd) naar de bedienapplicatie

Fout scenario's

- 1.

Relaties met andere use cases

Het feit dat de Trdl-verhinding is aangebracht wordt ook gerapporteerd aan de afnemende applicaties. Zie hiervoor de use cases m.b.t. "Melding van de actuele toestand".

De verhinding kan verwijderd worden middels use case TSR: Toestaan rijweginstelling.

De use case van VHR-E gaat op dezelfde wijze (niet uitgewerkt).

Alternatieve scenario's

2. -

Trdl-verhinderingen (voor gedetailleerdere eisen zie par. 3.3.1).

UC-VHR	Astris moet aan bedienapplicaties de mogelijkheid bieden om (meerdere) Trdl-verhinderingen voor rijweginstelling aan te brengen op seinen, secties, wissels en kruisingen.
UC-VHB	Astris moet aan bedienapplicaties de mogelijkheid bieden om (meerdere) Trdl-verhinderingen voor bediening aan te brengen op bedienbare infraelementen en gebieden.
UC-TSR	Astris moet aan bedienapplicaties de mogelijkheid bieden om aangebrachte Trdl-verhinderingen voor rijweginstelling weer te verwijderen.
UC-TSB	Astris moet aan bedienapplicaties de mogelijkheid bieden om aangebrachte Trdl-verhinderingen voor bediening weer te verwijderen. De gedetailleerde eisen bevatten een uitgebreidere beschrijving (1 a 2 pagina's) van de eis. Bijvoorbeeld van de Use Case Verhindere bediening (UC-VHB) namelijk VHB-1 en VHB-2 <i>Verhindere bediening</i>
VHB-1	Astris dient VHB-opdrachten te kunnen verwerken op bedienbare elementen of gebieden.
VHB-2	Astris dient er voor te zorgen dat BEVNL een TRDL bedienverhinderer actief maakt voor elk object (element of gebied) waarvoor in Astris één of meerdere (gestapelde) verhinderingen actief zijn. <i>Toelichting; Astris dient te voorkomen dat er een discrepantie ontstaat tussen de Trdl-VHB administraties van Astris en die van BEVNL. Als oplossing hiervoor zal Astris de eerste bedienverhinderer op een object pas accepteren, nadat BEVNL deze heeft geaccepteerd. Dit geldt uiteraard alleen voor objecten die in de beveiliging bekend zijn.</i>

Gedetailleerde functionele beschrijving 'Verhinderer voor rijweginstelling (VHR)'

Alleen secties, seinen, wissels en kruisingen kunnen verhinderd worden voor rijweginstelling.

Sein: Als een sein als beginsein is verhinderd voor rijweginstelling, dan mogen er geen rijwegen worden ingesteld vanaf dat sein. Analooq hieraan mogen er geen rijwegen worden ingesteld naar een sein dat als eindsein verhinderd is voor rijweginstelling. Een sein kan tegeliktijd als beginsein en als eindsein ("totaal") verhinderd zijn voor rijweginstelling dan zijn er helemaal geen rijweginstellingen meer mogelijk vanaf of naar dat sein.

Wissel/kruising: Als een wissel of kruis links is verhinderd voor rijweginstelling, dan mogen er geen rijwegen over de linkertak van dat wissel/kruis worden ingesteld. Analooq voor de rechtertak. Over een wissel/kruis dat zowel links als rechts ("totaal") verhinderd is voor rijweginstelling mogen helemaal geen rijwegen meer worden ingesteld.

Sectie: Als een sectie is verhinderd voor rijweginstelling mogen er helemaal geen rijwegen over worden ingesteld.

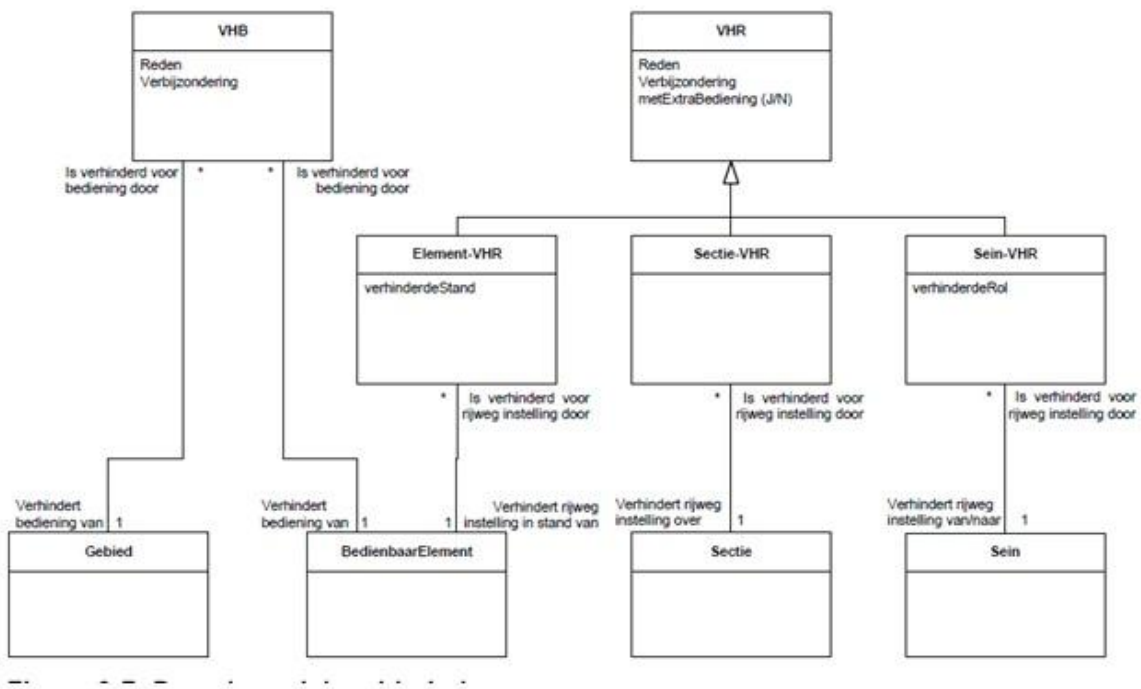
Merk op dat een VHR alleen invloed heeft op rijweginstellingen, en dus niet op andere functionaliteit (individuele bediening, maatregelen t.b.v. gebieden, etc.).

Als er een verhinderer voor rijweginstelling is aangebracht op een bedienbaar element dat gekoppeld is aan een ander bedienbaar element, dan heeft dat *geen* effect op de gekoppelde elementdeel.

System/Subsystem Design Description

De System/Subsystem Design Description (SSDD) document is wat betreft structuur weer vergelijkbaar opgebouwd als de SSS. Echter dit document gaat in op het ontwerp. Het beschrijft ontwerpbeslissingen, uitgangspunten en besluiten. Daarnaast bevat het domeinmodellen zoals het domein model 'Treindienstleider verhindere', zie figuur 61. Maar ook een architectuurmodel van het systeem, hardware architectuur, functionele use cases en het interface ontwerp.

Treindienstleider verhinderingen



Figuur 61 'Domeinmodel'. Bron: ASTRIS-SSDD.

SRS beveiligingscomponent

Het SRS is een Software Requirements Specification. Het beschrijft de eisen van de in de SSDD geïdentificeerde component. Volgens een vaste opbouw die bestaat uit een 'description', 'implementation', en 'rationale' worden de eisen in dit document beschreven. Ook worden attributen en parameters beschreven. Een eis wordt geïdentificeerd door een nummer. De eis wordt gedefinieerd door middel van een Description. In het Implementation veld wordt aangegeven door welke services van andere componenten de eis moet worden geïmplementeerd. Het waarom van de eis wordt toegelicht in het Rationale veld. Bij eisen die gesteld worden aan data staan tabellen met attributen die vastgelegd moeten worden. In deze tabellen zijn de attributen die de sleutel vormen van zo'n tabel **vet** gemarkeerd (Dit kunnen één of meerdere attributen zijn).

In de SRS wordt bijvoorbeeld verwezen naar de identificatie van de elementservice operatie 'Verhindert rijweginstelling'.

3.6.3.2 Element Service

De Elementen Manager Component biedt de element service aan. In de SSDD Astris (ISS00 ASTRIS) zijn in deze service de volgende elementservice operaties geïdentificeerd:

- Bedienen element (hieronder vallen alle bedienopdrachten op elementen (seinverlichting, grendel, rijrichtingkering (ook ULS en HLS) etc.)
- Verhindert bediening
- Toestaan bediening
- **Verhindert rijweginstelling**
- Toestaan rijweginstelling

De service operatie wordt in de SRS verder uitgewerkt.

In onderstaande tabel is van elke service operatie aangegeven van welke soort hij is, waar de definitie te vinden is, en in welke paragraaf de service operatie verder is uitgewerkt.

Naam	Soort ¹	Definitie	Uitwerking
BedienElement	Request/response	3.6.3.2.2	3.6.7.1
VerhinderBedienenElement	Request/response	3.6.3.2.3	3.6.7.2
ToestaanBedienenElement	Request/response	3.6.3.2.4	3.6.7.3
VerhinderRijwegElement	Request/response	3.6.3.2.5	3.6.7.4

Figuur 62 'Service operatie 'VerhinderRijwegElement'. Bron: ASTRIS-SRS.

Dat ziet er dan als volgt uit:

3.6.3.2.5 Interface eisen gesteld aan service request verhinderRijwegElement

Middels een verhinderRijwegElement service operatie verzoekt een consumer een element te verhinderen voor rijweg instelling. Voor bedienbare elementen moet worden aangegeven in welke stand de verandering geldt. Voor seinen moet worden aangegeven voor welke rol de verandering geldt.

Service Request	verhinderRijwegElement
ElementIdentificatie	ElementId van het voor rijweg te verhinderen element. Mogelijke element typen: <ul style="list-style-type: none"> o wissel o kruising o sein o sectie o bovenleidingsteller o ontspoortong
Stand [OPTIONEEL]	De stand van een bedienbaar element waarvoor de verandering geldt, verplicht als de verandering op een wissel of kruising wordt aangebracht. Mogelijke waarden: <ul style="list-style-type: none"> o links o rechts o totaal
Rol [OPTIONEEL]	De rol waarin het sein moet worden verhinderd, verplicht als de verandering op een sein wordt aangebracht. Mogelijke waarden: <ul style="list-style-type: none"> o begin o eind o totaal
Reden	De reden van de aan te brengen verandering. Mag alleen letters, cijfers, spaties en leestekens bevatten.
Verbijzondering [OPTIONEEL]	De verbijzondering van de reden van verandering voor rijweg. Mag alleen letters, cijfers, spaties en leestekens bevatten.
ExtraBediening [OPTIONEEL]	Geeft aan of de verandering met of zonder extra bediening is.
BedienaarType	Mogelijke waarden: <ul style="list-style-type: none"> o Centraal o Lokaal
OverdrachtNummer [OPTIONEEL]	Contractnummer om een ElementOpdracht uit te voeren welke overlap heeft met een gebied waar de bedienaar geen bedienbevoegdheid heeft.
OpdrachtgeverIdentificatie	Unieke instantienaam van de bedienapplicatie. Wordt door EMC alleen voor logging gebruikt.
OpdrachtgeverType	b.v. "TRDL" of "ARI". Wordt door de EMC alleen tbv logging gebruikt.
OpdrachtId	Unieke identificatie van de opdracht.

Eis: EMC-IF-09

Description	Een consumer dient een verandering voor rijweg op een element aan te kunnen brengen middels een verhinderRijwegElement service request.
Implementation	Intern
Rationale	

Figuur 63 'Definitie verhinderRijwegElement'. Bron: ASTRIS-SRS.

3.6.7.4 Verwerken ver hinderRijwegElement service request

In Tabel 3-22 staan de activiteiten opgesomd die uitgevoerd moeten worden naar aanleiding van een ver hinderRijwegElement service request.

Activiteit	Paragraaf	Exceptie Scenario
Bewaar ElementOpdracht	3.6.12.6	Exceptie 1
Controleer de combinatie van parameters in de opdracht	3.6.12.5	Exceptie 2
Controleer of dit element Rijwegverhinderbaar is	3.6.12.45	Exceptie 2
Controleer of ver hinderrol past bij seinattribuuttype (alleen voor seinen)	3.6.12.54	Exceptie 2
Lock element	3.6.12.8	Exceptie 2
Controleer bedienbevoegdheid op het element (hier voor wordt de interne service operatie controleerBedienbevoegdheid gebruikt)	3.6.9.3	Exceptie 3 indien NOK Exceptie 4 indien ten onrechte overdrachtsnummer meegegeven
Bedienbevoegdheid van het gekoppelde element dient niet bepaald te worden.		
Neem element in gebruik voor ver hindering voor rijweg (hiervoor wordt de service operatie NeemElementInGebruik gebruikt. Haal eerst de actuele stand van het element op)	3.6.9.4	Exceptie 3
Controleer uniciteit reden ver hindering voor rijweg	3.6.12.23	Exceptie 3
Indien één van de voorgaande controles niet geslaagd is: voer nevenstaand exceptie-scenario uit		Exceptie 5
Wijzig ElementOpdracht toestand (Uitvoeren)	3.6.12.11	Exceptie 2
Opslaan ver hindering voor rijweg	3.6.12.24	Exceptie 2
Geef element weer vrij voor ver hinderen voor rijweg. Hiervoor wordt de service operatie geefElementGebruikVrij gebruikt.	3.6.9.5	Exceptie 2
Unlock element	3.6.12.12	-
Wijzig ElementOpdracht toestand (Gereed)	3.6.12.11	-
Verstuur Response	3.6.12.13	

Tabel 3-22: Verwerken ver hinderRijwegElement Service Request

Alternatieven

Er zijn geen alternatieve scenario's.

Exceptie scenario's

In Tabel 3-23 staan de exceptie scenario's voor een ver hinderRijwegElement service operatie uitgewerkt.

Exceptie	Activiteit	Paragraaf
Exceptie 1	Verstuur Response zonder ElementOpdracht	3.6.12.18

Exceptie	Activiteit	Paragraaf
	Als er al een ElementOpdracht is: Wijzig ElementOpdracht (NOK/gereed), met daarin de NOKs die bij de activiteiten gedetecteerd zijn (als in de eis niet het wijzigen van de ElementOpdracht niet verwoord is)	3.6.12.11
	Verstuur Response met de geregistreerde NOKs (maar alleen als er nog geen response verstuurd is)	3.6.12.13
Exceptie 3	Voeg aan de ElementOpdracht de NOK toe die is geconstateerd. Ga door met de controles.	-
Exceptie 4	Voeg aan ElementOpdracht toe NOK/gereed, met reden 'ten onrecht overdrachtsnummer meegegeven. Ga door met controles	-
Exceptie 5	Unlock elementen	3.6.12.12
	Geef element weer vrij voor ver hinderen voor rijweg. Hiervoor wordt de service operatie geefElementGebruikVrij gebruikt.	3.6.9.5
	Wijzig ElementOpdracht (NOK/gereed).	3.6.12.11
	Verstuur Response met de in de ElementOpdracht geregistreerde NOKs.	3.6.12.13

Tabel 3-23: Exceptie scenario's ver hinderRijwegElement Service Operatie

Eis: EMC-FUNC-004	
Description	Een ver hinderRijwegElement service request dient verwerkt te worden zoals is gespecificeerd in Tabel 3-22: Verwerken ver hinderRijwegElement Service Request.
Implementation	Intern
Rationale	De activiteiten, alternatieven en exceptie scenario's zoals beschreven vormen gezamenlijk de afhandeling van een ver hinderRijwegElement service request.

Verwerken toestaanRijwegElement service request

Figuur 64 'Uitwerking ver hinderRijwegElement service request'. Bron: STD project ASTRIS.

Bijlage II BBMS Requirements

Het Pakket van Eisen van project BBMS vormt het requirements document dat alle requirements bevat. Dit zijn zowel functional als non-functional requirements. Dit alles in één grote tabel beginnend met de functional requirements en eindigend met non-functional requirements. Per requirement is vervolgens de MoSCoW methode toegepast. Dit houdt in een gradatie van Must have, Should have, Could have en Would have. Onderstaande volgt dit overzicht uit het Pakket van Eisen. Aansluitend treft u een voorbeeld van de inhoud van een Functioneel Ontwerp uit het project BBMS.

Algemene eisen

0.1	BBMS dient toegankelijk te zijn voor alle marktpartijen, waarbij middels autorisaties wordt vastgelegd welke functionaliteit en welke meetdatasets toegankelijk zijn. De eisen met betrekking tot autorisaties zijn beschreven in Error! Reference source not found. Error! Reference source not found.	M
0.2	BBMS dient toegerust te zijn op het opslaan en bewerken van minimaal de volgende data typen. L-PM Enkelvoudige punt metingen op lineaire assets vb: ultrasoon gebrek L-CM (semi-) continue meting op lineaire asset vb: geometrie meting L-VB (semi-) video beelden op lineaire asset vb: videoschouw L-FB Fotobeeld op lineaire asset vb: foto van spoorstaaf L-DM Documenten met meetwaarden van een lineaire asset vb: ultrasoon rapport P-PM Enkelvoudige punt metingen op punt object vb: kwaliteit ES-las P-CM (semi-) continue meting op punt object vb: poss meting P-FB Fotobeeld van punt object vb: foto van wisseltong P-DM Documenten met meetwaarden van een punt object vb: schouw rapport De codering wordt gebruikt om specifieke eisen toe te koppelen aan de betreffende meetdatasets. Voor de volledigheid wordt voor eisen die op alle datatypen van toepassing zijn onderstaande codering gebruikt: ALL Op alle datatypen van toepassing	M
0.3	BBMS dient schaalbaar en uitbreidbaar te zijn voor nieuwe typen meet- en metadata.	M
0.4	BBMS dient schaalbaar en uitbreidbaar te zijn voor nieuwe functionaliteit op de gebieden: beheren stamdata, opnemen meet- en metadata, analyseren en ontsluiten.	M
0.5	BBMS is voorzien van een gebruiksprotocol waarin aangegeven is welke waarde aan de gegevens in BBMS kan worden toegekend en waar de gegevens voor gebruikt kunnen worden. Deze toelichting is als geheel oproepbaar in de help-omgeving. Daarnaast kan per parameter van een meetdataset specifieke toelichting opgeroepen worden in de analysetool zelf.	S
0.6	Het systeem is voorzien van twee webportalen, waarmee gegevens- ontvangst en ontsluiting plaatsvindt.	C
0.7	Data in BBMS (autorisaties / assets / normen / meetdata) zijn voorzien van een geldigheidstermijn die kan worden aangepast per norm, parameter van een meetdataset of asset.	S
0.8	Meetdata in BBMS is altijd voorzien van een datum/tijd indicatie die het moment van de meting aangeeft en van locatie informatie die nauwkeurig de plaats van de meting aangeeft in een algemeen bekend en toepasselijk coördinatenstelsel.	M
0.9	Werken met periodes (van-tot) is binnen BBMS mogelijk en deze is door de gebruiker vrij instelbaar. BBMS selecteert vervolgens alleen geldige assets, normen en meetdata binnen deze periode.	M
0.10	BBMS kent de mogelijkheid van benchmarking. Met benchmarking wordt in dit kader bedoeld het vergelijken van meetdatasets binnen specifieke geocodes, contractgebieden, baanvakken en sporen ten opzichte van het gemiddelde van de betreffende meetdataset over een grotere selectie of over heel Nederland.	C

0.11	Marktpartijen kunnen binnen BBMS in een speciaal voor hen (geautoriseerd) deel van BBMS hun eigen analyses, rapportages, views en exports samenstellen. Ze kunnen deze geheel zelf samenstellen, maar ook analyses, rapportages, views en exports die voor alle gebruikers beschikbaar zijn, of die ze zelf eerder hebben samengesteld, op eenvoudige wijze kopiëren en aanpassen.	M
0.12	BBMS beschikt over back-up faciliteiten en archiveringsmogelijkheden voor zowel het volledige systeem als geselecteerde meetdatasets. Dit kan handmatig gedaan worden en ingesteld worden tot een automatisch repeterend proces.	M
0.13	Zowel de gebruikers- als de beheerdersdocumentatie dient in het Nederlands opgesteld te zijn.	S
0.14	Binnen het gehele BBMS pakket (portals, analysetool, etc.) worden automatisch notificaties verstuurd. De beheerder kan de inhoud van de notificatie configureren (vaste tekst, parameters, etc.), de te adresseren gebruikersgroepen specificeren, en de voorwaarden definiëren wanneer een notificatie verstuurd zal worden (processtappen, databaseacties, etc.).	C

Beheren stamdata

1.1.1	De unieke definitie van assets (usis ++) waarmee meetdata wordt geregistreerd is aanwezig c.q. kan worden vastgelegd. (zie ook bijlage 2)	M
1.1.3	BBMS kan data vanuit de bronbestanden waarin het beheer van de assets plaatsvindt (SAP, BBK en Infra Atlas) importeren. Deze data is voorzien van USIS identificatie. Een releasemanagement proces, inclusief versie beheer op asset datasets, is hiervoor aanwezig. Binnen dit proces wordt gewerkt met een importschema en een autorisatie schema.	M
1.1.4	Bij iedere release wordt aan BBMS een actuele asset dataset aangeleverd. Door een vergelijking van de asset datasets (actuele en vorige release) is de delta in de datasets beschikbaar. Deze delta wordt inzichtelijk gemaakt en getoond aan de eindgebruiker. (actuele lijst = basislijst + mutaties) Tevens wordt de delta gebruikt voor het actualiseren van meetdatasets.	S
1.1.5	Een release van asset data ten behoeve van BBMS kan volledig of deels worden uitgevoerd. BBMS kan deelreleases verwerken en laten passen op de bestaande asset data. De delta wordt inzichtelijk gemaakt en getoond aan de eindgebruiker. Voor deelreleases zijn selecties mogelijk op: <ul style="list-style-type: none"> • asset type c.q. asset groep • contract gebied • geocode gebied • spoornummer • emplacement • wisselnummer • kilometrering 	S
1.1.6	BBMS sluit aan op het releasebeleid USIS (assetdataset) en kan omgaan met meerdere naast elkaar bestaande USIS versies.	M
Beheer Normen (1.2)		
1.2.1	Voor een meetdataset kunnen normen worden vastgelegd. Het betreft hier de normen zoals deze zijn vastgelegd in de ProRail richtlijnen voor instandhouding. Aan elke parameter van een meetdataset kunnen normwaarden worden gekoppeld zodat overschrijdingsrapportages en analyses plaats kunnen vinden.	M
1.2.2	Invoer, wijzigen en verwijderen van normen is mogelijk. Een logging mechanisme is aanwezig zodat bekend is welke verandering, wanneer en door wie is uitgevoerd. Vorige versies van normen blijven bewaard en kunnen worden gebruikt voor analyses.	M
1.2.3	Normen worden vastgelegd met een bepaalde geldigheidstermijn en verwijzing naar het bijbehorende normenkader.	S
1.2.4	Normen kunnen in elektronisch formaat (XML) worden aangeleverd. Gewerkt wordt met een mechanisme van aanbieden en vrijgeven waarbij de normen door de ene afdeling van ProRail worden aangeboden en door een andere afdeling wordt vrijgegeven binnen BBMS	C
1.2.5	Behalve normen kunnen meettoleranties worden vastgelegd. Meettoleranties kunnen op eenzelfde wijze als de normen worden beheerd.	M
Beheer autorisaties (1.3)		
De eisen met betrekking tot autorisaties zijn beschreven in hoofdstuk 9.		

	Onderhoud meet & meta data (1.4)	
1.4.1	Wanneer mutaties in USIS van invloed zijn op de wijze waarop de meet- en meta data aan USIS is gekoppeld, vindt actualiseren van meet- en meta data bij ontvangst en/of release van een nieuwe versie van de Asset dataset plaats.	S
1.4.2	Invoer, wijzigen en verwijderen van meet- en meta data is geautoriseerd mogelijk. De beheerder kan per attribuut bepalen of dit gewijzigd mag worden. Er is een logging mechanisme aanwezig zodat bekend is welke verandering, wanneer en door wie is uitgevoerd.	S
1.4.3	Per parameter uit een meetdataset kan een vrijgave plaatsvinden	S
1.4.4	De gespecificeerde, gewijzigde en nieuw toe te voegen meetdatasets kunnen eenvoudig worden geconfigureerd. Het pakket dient overweg te kunnen met RD en GPS coördinaten en met spatial datatypes die voorkomen in de laatste versies van GIS pakketten. Dit is standaardfunctionaliteit van het pakket.	M
1.4.5	In BBMS is het mogelijk om historie, status en voortgang van aanleveringen van meet- en metadatasets in te zien en te bewaken.	S
1.4.6	De meetdatasets en bijbehorende metadata, normen, en bepaalde standaardrapportages en analyses worden minimaal 10 jaar na het jaar van inwinnen bewaard worden in BBMS.	
1.4.7	Alle historische gegevens zijn direct beschikbaar voor analyse, weergave en export.	

Opnemen meetdata

2.1.1	Per mail worden notificaties uitgestuurd conform een vooraf opgesteld distributie schema bij het actualiseren van de asset configuratie data.	C
2.1.2	Om inzicht te houden in de te verwachten meetdata en de realisatie van de gecontracteerde meetdiensten wordt op de asset data in verschillende lagen de dekking aangegeven voor de op te nemen meetdata. Deze lagen zijn: Contract Planning Uitvoering Oplevering meetgegevens Dekking aanbrengen houdt in dit verband in het per meetdataset of parameter van een meetdataset aan een laag toewijzen van delen van de railinfrastructuur. BBMS is ingericht om op de asset data in 4 lagen dekking aan te kunnen brengen	C
2.1.3	Daar het contract kan bestaan uit meerdere metingen per jaar, wordt op de dekkingslaag contract de frequentie aangegeven. Voor de dekkingslagen planning, uitvoering en meetgegevens zijn per laag meerdere dekkingen per jaar beschikbaar.	C
2.1.4	Dekkingslagen kunnen zowel handmatig aangebracht worden in BBMS zelf als geïmporteerd uit een externe bron.	C
2.1.5	Het handmatig vullen van een laag wordt gedaan op grafische kaarten door het aanklikken van de delen van de railinfrastructuur (per USIS-id of alle USIS-id's binnen een door de gebruiker aan te duiden bereik), waarmee dat element toegewezen wordt aan een van de dekkingslagen.	C
2.1.6	Dekkingslagen kunnen eenvoudig toegewezen worden aan individuele parameters, aan een meetdataset of aan meerdere meetdatasets tegelijk.	C
2.1.7	Gebruikers kunnen vergelijkingen maken tussen verschillende lagen waarbij rekening gehouden wordt met de frequentie per jaar en de meerdere dekkingen per laag. Hiermee is BBMS in staat te beoordelen of bijvoorbeeld een spoor met frequentie 3x per jaar in de contractlaag afgedekt wordt door 3 dekkingen in planning, uitvoering of meetgegevens.	C
2.1.8	De verschillen in dekking worden zowel tabellarisch als grafisch weergegeven.	C
2.1.9	De dekkingslagen kennen binnen BBMS gelijke beheers- en gebruiksfuncties als de meet- en metadata, zoals autorisaties, instellen tijdsperiode, exports etc.	C
2.1.10	Beschikbaar stellen (vrijgeven) van nieuwe asset configuratie data release aan gebruikers (pull mechanisme)	C
2.1.11	Ophalen van (een subset van) de meest recente asset dataset gegevens door de eindgebruiker (pull). Hierbij wordt rekening gehouden met het autorisatieprofiel van de eindgebruiker en zijn onderstaande selecties mogelijk: <ul style="list-style-type: none"> asset type c.q. assetgroep contract gebied geocode gebied 	C

	<ul style="list-style-type: none"> • spoornummer • emplacement • wisselnummer • kilometreering 	
2.1.12	Informeren van eindgebruikers over nieuwe en actuele status van asset datasets.	C
	Ontvangen meet & meta data (2.2)	
2.2.1	Bij het innemen van meetdata wordt bepaald in hoeverre de afspraken uit het meetcontract al zijn nagekomen. De planning van de te meten baanvakken en sporen wordt daarbij vergeleken met de sporen waarvan meetgegevens zijn ingenomen. Voor sporen waar “gaten” in de meetdata zijn ontstaan wordt de dekkingsgraad aangegeven (het aantal en omvang van de “gaten”)	S
2.2.3	Bij het ontvangen van meet- en metadata wordt vastgelegd wie de leverancier is van de data	W
2.2.4	De leverancier van meet- en metadata levert zijn gegevens via een standaard mechanisme zelf aan en zorgt ervoor dat deze tijdig op een vooraf afgestemde locatie in het (ProRail) netwerk gereed staat voor opname binnen BBMS. BBMS importeert de gegevens en slaat deze op in de BBMS-database	M
2.2.8	Om na het importeren meetdata zo nauwkeurig mogelijk te kunnen vergelijken is het in BBMS mogelijk om meetdata na importeren nog te bewerken voor wat betreft hun positionering. De mogelijke bewerkingen op individuele meetdata bestaan uit verschuiven, rekken, krimpen, knippen en plakken.	S
2.2.9	<u>Het is mogelijk meetdata te verschuiven.</u> Dit houdt in het toewijzen van een nieuwe startlocatie voor een geselecteerd deel van de datareeks, waarna automatisch alle datapunten eenzelfde verschuiving qua positionering krijgen.	S
2.2.10	<u>Het is mogelijk meetdata te rekken en krimpen.</u> Dit houdt in het toewijzen van een nieuwe startlocatie voor het gedeelte, waarna automatisch alle datapunten qua tussenafstand gelijkmatig gepositioneerd worden tussen deze locaties. Hierbij zijn grenzen ingesteld voor hoeveel de resulterende tussenafstand mag afwijken van de originele tussenafstand	S
2.2.11	Bij het rekken en krimpen kunnen ook punten waarvan de exacte positie bekend is in RD of GPS coördinaten als start en of eindlocatie worden gekozen. Dit is een automatisch proces waarbij handmatige correctie mogelijk is door de gebruiker.	C
2.2.12	<u>Het is mogelijk meetdata te knippen en plakken.</u> Dit houdt in het uitsnijden van een gedeelte van de datareeks en het aan andere USIS-id's toewijzen van dit gedeelte.	S
2.2.13	Bij de bewerkingen zijn steeds de naam, labels (geocode, spoornaam etc) en grenzen beschikbaar van de asset waarop het gedeelte van de meetdata zich bevindt, zodat een gedeelte van de meetdata eenvoudig toe te wijzen of te bewerken is in relatie tot de bijbehorende asset.	C
2.2.14	Op de datareeksen van meerdere metingen van een gedeelte van de infra kan een automatische onderlinge uitlijning worden uitgevoerd. Op basis van het overeenkomen van meetwaarden tussen de datareeksen wordt elke datareeks individueel uitgelijnd. De gebruiker kan de automatische wijzigingen accepteren of afwijzen en heeft ook de mogelijkheid om handmatig uit te lijnen.	S
2.2.15	Bij het uitlijnen is instelbaar of en welke meetdata als referentie dient. Deze meetdata mag niet verschuiven tijdens het uitlijnen.	S
2.2.16	Als door het verschuiven, rekken of krimpen of vergelijken meetdata over de grens van een asset heen geschoven wordt, wordt naar keuze van de gebruiker automatisch de aansluitende meetdata uit dezelfde meting ook opgeschoven of het gedeelte dat niet meer toegewezen kan worden aan de asset geïsoleerd weergegeven voor verdere bewerking.	S
2.2.17	Gewerkt kan worden met Key Performance Indicatoren (KPI's). KPI's kunnen worden vastgelegd en KPI rapportages kunnen op eenvoudige wijze worden samengesteld. KPI's kunnen automatisch periodiek worden samengesteld en verstuurd naar beheerders. Voorbeelden: - Percentage tijdig aangeleverde meetdatasets per meetbedrijf - Gemiddelde tijd tussen aanleveren meetdata en vrijgave meetdata - Hoeveelheid afkeur van meetdatasets	C
	Controle meet & meta data (2.3)	
2.3.1	Per aangeleverde meetdataset uitvoeren van controle op volledigheid en kwaliteit van de meetdata. De wijze waarop dit plaatsvindt is	M

	afhankelijk van de meetdataset en is bij het specificeren van deze set opgenomen. Controle dient onder meer plaats te kunnen vinden per corridor, baanvak en spoor en kan zowel automatisch als handmatig plaatsvinden.	
2.3.2	Afdrukken controle rapporten waarin melding wordt gedaan van: <ul style="list-style-type: none"> - tijdigheid van aanlevering - actualiteit van de meetdata - volledigheid van aangeleverde meetdata - juistheid van aangeleverde meetdata (zie bijlagen per meetdataset). 	S
	Vrijgave meet & meta data (2.4)	S
2.4.1	Indien controle op de meetdata succesvol heeft plaatsgevonden, wordt de meetdata set aangeboden voor vrijgave. Het vrijgave proces voorziet in: <ul style="list-style-type: none"> - Opnemen van de correcte meetdata in BBMS - Afwijzen van incorrecte meetdata - Vastleggen en afdrukken van een vrijgave overzicht met daarop informatie over het vrijgave proces (metagegevens) en een overzicht van afgewezen meetdata. 	M
2.4.2	Op basis van de autorisatie op meetdatasets, worden eindgebruikers bij vrijgave van een nieuwe meetdataset, automatisch per mail geïnformeerd over de bijgewerkte status van de meetdata. De voorwaarden dienen nader te worden bepaald om tot een adequate vorm van berichtgeving te komen.	C

Analyseren meetdata

	Analyseren meetdata (3.1)	M
3.1.1	Het BBMS biedt analyse functionaliteit voor onderzoeksdoeleinden. Hierbij kunnen meerdere parameters van meetdatasets worden geselecteerd en worden gecombineerd. Analyses op deze bestanden kunnen worden gedefinieerd en uitgevoerd. De resultaten kunnen worden gepresenteerd in grafiekvorm. Zowel het samenstellen van eigen analyses als het uitvoeren van vooraf gedefinieerde analyses is mogelijk. Het type en het ontwerp van de database dient dit mogelijk te maken. Minimaal de volgende functionaliteiten zijn beschikbaar: <ul style="list-style-type: none"> Classificaties Wiskundige bewerkingen Statistische bewerkingen Filters (type en dimensies vrij instelbaar) Logische functies Database functies De leverancier levert bij de aanbidding een lijst met bewerkingen en functies die in het standaardpakket beschikbaar zijn.	S
3.1.2	De analyse functionaliteit kan worden toegepast op alle meetdatasets.	S
3.1.3	Bij het analyseren van meetdata kan worden aangegeven welke (subset van de) meetdata en welke versie van de normen gebruikt wordt.	C
3.1.4	Door de <u>normen</u> op meetwaarden toe te passen vindt classificatie plaats. De classificaties die per meetdataset van toepassing zijn worden bij die meetdataset beschreven. Classificaties worden gebruikt binnen rapportages, views en exports.	S
3.1.5	Het vergelijken van meerdere normen (verschillende versies) ten opzichte van elkaar is bij het analyseren van meetdata mogelijk	S
3.1.6	geanalyseerde meetdata kan worden opgeslagen (indien verrijking heeft plaatsgevonden) en zijn voor gebruikers toegankelijk middels rapportages, views en exports.	S
3.1.7	BBMS voorziet naast analyse van meetdata in de mogelijkheid om geografische analyses uit te voeren. De resultaten worden gepresenteerd in de standaard GIS viewer van ProRail. Als aangetoond kan worden dat, gezien over de hele levenscyclus van de applicatie, een andere GIS viewer voordelen heeft is in overleg met ProRail een andere oplossing mogelijk.	S

3.1.8	Voor het beheer van het meetdienstencontract is het mogelijke direct inzicht in te krijgen in de voortgang van uitvoering van een campagne of een contract. Zowel via de GIS-viewer als middels rapportages.	S
-------	--	---

Ontsluiten meetdata

	Rapportages samenstellen en beschikbaar stellen (4.1)	
4.1.1	Per meetdataset wordt een set van standaardrapportages binnen BBMS beschikbaar gesteld. De specificatie per meetdataset benoemt deze rapporten.	S
4.1.2	De eindgebruiker heeft de mogelijkheid om zelf (ad hoc) rapportages samen te stellen binnen een meetdataset (enkelvoudige rapportages).	S
4.1.3	De eindgebruiker heeft de mogelijkheid om zelf (ad hoc) rapportages samen te stellen waarbij gegevens vanuit meerdere datasets betrokken zijn (meervoudige rapportages).	S
4.1.4	De eindgebruiker kan rapportages opstellen vanuit de viewers en vanuit de analysetool.	C
	Beschikbaar stellen views (4.2)	
4.2.1	Per meetdataset wordt een set van standaard views binnen BBMS beschikbaar gesteld. De specificatie per meetdataset benoemt deze views. Er zijn verschillende typen viewers. - Een GIS viewer om de assets (bijv. een stuk spoor of een wissel) in coördinaatstelsel RD en GPS op de kaart van Nederland weer te geven. - Een meetdata viewer om de ruwe meetdata en geanalyseerde meetdata te tonen en - Een grafische viewer voor foto en videomateriaal die voldoet aan de ProRail aansluitvoorwaarden [2]. ProRail heeft daarbij een voorkeur voor Silverlight.	S
4.2.2	Het portaal voorziet in het synchroniseren van één of meer datasets in alle viewers. Synchronisatie vindt plaats onafhankelijk van de viewer waarin genavigeerd wordt. Dit werkt zowel voor puntobjecten als voor (de positie binnen) strekobjecten. Synchronisatie is instelbaar als: Uit, Op aanvraag (via knop of functietoets) en Continu.	S
4.2.3	De eindgebruiker heeft de mogelijkheid om zelf (ad hoc) enkelvoudige views samen te stellen binnen een meetdataset.	S
4.2.4	De eindgebruiker heeft de mogelijkheid om zelf (ad hoc) meerdere views van dezelfde of verschillende meetdatasets naast elkaar op te roepen.	S
4.2.5	Het systeem is zowel vanuit de GIS viewer als vanuit de meetdata viewer bereikbaar. Het is mogelijk om vanuit de meetdata viewer direct naar hetzelfde punt in de GIS viewer te gaan en weer terug.	C
4.2.6	Gebruikers kunnen de door hen ingestelde analyses, rapportages en views opslaan, zodat deze bij een volgende keer inloggen beschikbaar zijn. Analyses, rapportages en views van een gebruiker kunnen aan andere gebruikers ter beschikking worden gesteld.	S

	De GIS viewer	
4.2.6a	<p>De GIS viewer dient verschillende kaartlagen in de vorm van web services kunnen ontsluiten. Toegestane services zijn ESRI Mapservice of WMS. De geografische kaartlagen zijn:</p> <ul style="list-style-type: none"> - Basis geografische gegevens: BBK/ USIS - Oriëntatiegegevens <ul style="list-style-type: none"> o Luchtfoto's langs het spoor o Land dekkende luchtfoto's/Top10nl - De Geocode gebieden en de contractgebieden van de aannemers. <p>De genoemde kaartlagen komen alle voor in Railmaps, de huidige GIS viewer.</p>	S
4.2.7	De GIS viewer is intuïtief en bevordert het gebruik van het systeem voor minder ervaren eindgebruikers.	S
4.2.8	<p>De GIS viewer dient over de onderstaande functionaliteiten te beschikken:</p> <ul style="list-style-type: none"> - Selecteren <ul style="list-style-type: none"> o Van een of meerdere asset typen o Van een of meerdere assets - Navigeren <ul style="list-style-type: none"> o Play/Stop/Still o Forward o Reverse o GoTo, spring naar een specifiek opgegeven, bijv. kilometrering o In- en uitzoomen o Pannen - Zoeken <ul style="list-style-type: none"> o USIS ID o Op Geocode o Op Geocode + spoornummer o Op Geocode + spoornummer + kilometer (lint plus km) o Op contractgebied o Op corridor, op baanvak o Op combinaties hiervan o Op een door een gebruiker gedefinieerde indeling 	S
4.2.9	<p>De gebruiker heeft binnen de GIS viewer de mogelijkheid om het assettype te selecteren waarmee gewerkt wordt (baan / bovenleiding). Na deze selectie worden alle assets vallend onder het asset type in de GIS viewer gepresenteerd. Het gelijktijdig selecteren van assets vallend onder meerdere asset types is mogelijk. Gebruikers kunnen deze selectie bewaren voor gebruik in een volgende sessie.</p>	S
4.2.10	<p>Nadat een asset (type) is geselecteerd kan de gebruiker specifieke assets die behoren bij de asset type selecteren. De gebruiker ontvangt een overzicht van de aanwezige meetdatasets (views) en hij selecteren welke view hij wil openen. Het maximaal aantal gelijktijdig te openen views kan per gebruiksgroep worden ingesteld.</p>	s
4.2.11	De gebruiker heeft vanuit de GIS viewer de mogelijkheid om door geopende meetdatasets (views) te lopen middels navigatie mogelijkheden.	S
4.2.12	De gebruiker heeft vanuit de GIS viewer de mogelijkheid om een bepaalde asset op te zoeken.	S
4.2.13	De eindgebruiker heeft de mogelijkheid om geografische selecties op meetdatasets uit te voeren met behulp van muis.	S

4.2.14	De eindgebruiker heeft de mogelijkheid om selecties op basis van alle attributen van meetdatasets uit te voeren, bijv. op geocode, contractgebied	S
4.2.15	De GIS viewer zorgt ervoor dat de huidige locatie altijd in beeld is, zo nodig wordt de extend van de kaart dus aangepast.	c
4.2.16	De gebruiker beschikt over een snelheidsknop waarmee hij aan kan geven met welke snelheid de meetdata wordt doorlopen.	S
4.2.17	Performance eisen: De GIS viewer dient de doorlopende meetdata met een instelbare snelheid te kunnen weergeven. De maximum snelheid dient configureerbaar en ten minste 200 km/uur te zijn. De engine dient de viewers met deze snelheid te kunnen synchroniseren.	S
	De meetdata viewer, algemene functies	S
4.2.18	De meetdata viewer dient over de onderstaande functionaliteiten te beschikken: - Instellingen - Selecteren - Navigeren - Zoeken	S
4.2.19	Meetdata die is vastgelegd in binnen BBMS bekende berichtformaten en met een in BBMS bekende locatieaanduiding dienen via BBMS ontsloten te kunnen worden.	S
4.2.20	BBMS heeft een gebruikersvriendelijke meetdata viewer voor gebruikers die het systeem intensief gebruiken voor (ad hoc) analyses en (ad hoc) rapportages op lineaire assets als punt objecten.	C
4.2.21	Vanuit de meetdata viewer moet het mogelijk zijn direct de geselecteerde assets te tonen in de GIS viewer.	S
4.2.22	Data moeten bewerkt kunnen worden met behulp van systeembrede en individuele configuraties, algoritmen en filters. Doel hiervan is het beoordelen van de data. Het moet mogelijk zijn weergaves, analysemethoden en objecten samen te stellen ten behoeve van verwerking. Vanuit de meetdata viewer dient de applicatie gestart en gestopt kunnen worden.	S
4.2.23	Eisen aan presentatie in vensters Configuratie Definiëren diagrammen, grafieken en rapporten (=analyses) Structuur Aan kunnen maken van mappen Definiëren snelkoppelingen naar geocodes, campagnes, contractgebieden Opslaan van zoekresultaten in mappen Weergave Kunnen presenteren van alle producten Kunnen kiezen uit lijst met laatst gebruikte producten	S
4.2.24	Analyses moeten gerangschikt kunnen worden in mappen en productcatalogi	S

4.2.25	<p>De volgende analyseproducten dienen beschikbaar te zijn:</p> <p>Diagrammen</p> <p>Grafisch weergeven van data (zie ook 4.2)</p> <p>Uitlijnen van afzonderlijke diagrammen op overeenkomende X-as</p> <p>Aanvullende eis: Uitlijnen op X-as (positie) op basis van patroonherkenning</p> <p>Combineren van data in één diagram met gelijke X en Y-as en omgekeerd (splitsen)</p> <p>Combineren en ordenen van diagrammen in een venster (al dan niet op basis van een thema)</p> <p>Voorgedefinieerde combinaties van diagrammen</p> <p>Omzetten van diagram naar tabel</p> <p>Tabellen</p> <p>Opmaken en ordenen van tabel (aanmaken en beheren van diverse standaardopmaken)</p> <p>Toevoegen omschrijving aan tabel</p> <p>Toevoegen van berekeningen aan tabel (gemiddelde, overschrijding, etc.)</p> <p>2-D weergave</p> <p>Gegevens weergeven op punt- en lijnobjecten in een vlak (op schaal of gestileerd)</p> <p>Configureren van de gewenste weergave van punten lijnstukken vlakken met kleuren, symbolen, lijntype, e.d. op basis van waarde, berekening eenvoudig accentueren, highlighten</p> <p>3-D weergave (geen verplichte eis/ should have)</p> <p>Weergave van spoorbaan en een aantal diagrammen in perspectief</p> <p>een 3-D indruk ontstaat.</p>	S
4.2.26	<p>Voor de diagrammen dienen de volgende presentatievormen beschikbaar te zijn:</p> <p>Lijnweergave</p> <p>Lijnkleur afhankelijk van waarde, lijntype, lijn met symbool, lijn met tekst (en alle combinaties)</p> <p>Vlakweergave</p> <p>Vlakkleur afhankelijk van waarde, vlakvulling hoogte 100%, vulling vanaf basis tot waarde, vulling vanaf grenswaarde tot actuele waarde, vlak tussen aangrenzende grenswaarden, vlak met tekst (en alle combinaties)</p> <p>Kolomweergave</p> <p>Kolomkleur afhankelijk van waarde, hoogte 100%, vanaf basis tot waarde, vanaf grenswaarde tot actuele waarde, met tekst, met symbool (en alle combinaties)</p> <p>Segmentweergave</p> <p>Segmentkleur afhankelijk van waarde, tekst in of naast segment, met symbool (en alle combinaties)</p> <p>Vastleggen voorkeur en alternatieve standaardconfiguraties per analyse</p> <p>Weergeven KPI cijfers tussen willekeurige tijdstippen en/ of in een willekeurig gebied</p> <p>Overlay venster</p> <p>Laat bijv. het gemiddelde over een willekeurig traject zien</p> <p>NB Deze eis is niet verplicht als hiervoor een goede analyse mogelijkheid beschikbaar is.</p>	S

4.2.27	De volgende werkwijzen dienen ondersteund te worden: Openen grafiekweergave en diagrammen Instellen tijdbereik Kiezen objecten multiselect mogelijk is en op verschillende niveaus gekozen kan worden Kiezen campagnes Alle genoemde functies moeten ook weer opgeheven kunnen worden.	S
4.2.28	Grafiekweergave dient zowel via menu als met klikken en slepen mogelijk te zijn. Er moet gekozen kunnen worden welke structuurelementen en geocodes weergegeven worden.	S
4.2.29	Het moet mogelijk zijn diagrammen toe te voegen, te verbergen, te verwijderen en wijzigingen ongedaan te maken Opmaak mogelijkheden diagram <ul style="list-style-type: none"> • positie wisselen • vergroten, verkleinen, rekken en strekken • opmaak aanpassen • naam, breedtefactor, lineair/ logaritmisch • keuze naam of eenheid, keuze van eenheid • tekst uitlijning en rotatie • bereik • schaal • hulplijnen 	S
4.2.30	Met gegevens uit grafiekweergaven (diagrammen) moeten exports gemaakt kunnen worden.	C
4.2.31	Het datumbereik voor een grafiekweergave moet beperkt kunnen worden tot een campagne, tijdvak, of datum	S
4.2.32	Eisen ten aanzien van het afdrukken van grafiekweergaven: Printerinstellingen, Afdrukvoorbeeld, Afdrukbereik (Alles, Huidig beeld, Pagina's)	S
4.2.33	Grafiekweergave moeten gekopieerd kunnen worden naar het Windows klemboard.	C
4.2.34	In plaats van met de schuifbalk moet het mogelijk zijn direct naar een ingevuld kilometrage binnen een kilometerlint te springen.	C
4.2.35	Bij grafiekweergave moeten vensters trapsgewijs, horizontaal en verticaal gerangschikt kunnen worden. Structuurelementen (geocode, kilometrage en campagne) van de ene grafiek moeten in de andere grafiek kunnen worden overgenomen. Het moet mogelijk zijn vensters, of combinatie van vensters te synchroniseren. Gekoppeld aan structuurelementen, met behoud van offset, met behoud van schaalverschil (zoomniveau).	S
4.2.36	Het moet mogelijk zijn te zoeken naar meetwaarden en combinaties van meetwaarden. Ook moet het mogelijk zijn om zoekwaarden te accentueren en vooruit en achteruit te navigeren naar resultaten.	C
4.2.37	Het moet mogelijk zijn opmerkingen toe te voegen en deze te configureren (wel of niet afdrukken, verplaatsen, lengte, breedte, navigeren)	C
4.2.38	In grafische weergaven moeten gebieden zijn aan te wijzen (af te grenzen). De gegevens binnen de aangegeven grenzen moeten geselecteerd kunnen worden en in tabelvorm in een rapportage geplakt kunnen worden.	C
4.2.39	In grafische weergaven moeten grenswaarden kunnen worden aangegeven. De gegevens boven onder of tussen de grenswaarden moeten geselecteerd kunnen worden en in tabelvorm in een rapportage geplakt kunnen	C
4.2.40	Het moet mogelijk zijn al dan niet bewerkte grafieken en bijbehorende opmerkingen te selecteren en deze op te slaan (werksessies). De opgeslagen grafiek kan dan afzonderlijk bewerkt worden.	C

4.2.41	<p>Het moet mogelijk zijn gedrag en weergave van de presentatievensters van BBMS aan te passen. Hieronder volgt een <i>niet uitputtende</i> lijst van gewenste instelmogelijkheden.</p> <ol style="list-style-type: none"> 1. Aan en uit zetten tooltips. Opmaak tooltips. 2. Tonen doelpositie bij slepen en neerzetten. 3. Bij afdrukken configureren van af te drukken (grafiek)onderdelen. 4. Wijze van laden van gegevens om het presenteren van gegevens (vooral grafisch) te kunnen versnellen. 5. Kliknauwkeurigheid muis. 6. Vergrotingsfactor bij in- en uitzoomen. 7. Tussenafstanden schaalverdeling. 8. Cijfers achter de komma (offsetwaarden). 9. Laatst gebruikte zoomniveau bewaren voor volgende sessie. 10. Keuze tussen vector of bitmap formaat bij kopiëren naar klembord. 11. Aanmaken bewaren, aanpassen en verwijderen van sjablonen. 12. Configureren van weergave van sjablonen. 13. Pagina instelling (zoals marges, formaat, kop- en voettekst. 14. Afdrukvoorbeeld. 15. Afdrukken legenda en metagegevens (bijvoorbeeld meetperiode). 16. Wijze en snelheid van navigatie door frames. 17. Doorzichtigheid en weergave van achtergrond en randen van 3D frames. 18. Schaal van y-as op basis van getoonde of alle waarden. 19. Weergave van tijdelijke opmerkingen (zoals kleur, lettertype, afmetingen, plaatsing. 20. Wachtwoord. 21. Map voor tijdelijke bestanden. 22. Cache voor configuratieobjecten. 23. Comprimeren van verkeer met server. 24. Verbinding met server online of offline. 25. Tonen van meldingen aan en uit kunnen zetten. 26. Beheren van een persoonlijke lijst met veelgebruikte baanvakken, campagnes, contractgebieden, etc. 27. Kopiëren van persoonlijke lijsten door andere gebruikers (naar keuze vervangen of toevoegen). 	S
4.2.42	<p>Foto's moeten geopend kunnen worden via links in grafieken en tabellen. Door foto's dient handmatig genavigeerd te kunnen worden, maar ook via een dia voorstelling met instelbare snelheid. Helderheid, contrast, grijswaarden moeten aangepast kunnen worden. Foto's en uitsneden daarvan moeten naar verschillende formaten geëxporteerd kunnen worden in verschillende resoluties.</p>	S
4.2.43	<p>Data-editor functies. Data moeten aangemaakt, bewerkt en verwijderd kunnen worden, ondersteund met sneltoetsen. Doel is om data in één bewerking in te voeren en daarna in één keer (dus sneller) in de database te laden (exporteren). Na het invoeren moeten data eenvoudig gekopieerd, gewijzigd en verwijderd kunnen worden. De ingevoerde data moeten bewaard kunnen</p>	S

4.2.44	Er dient contextgevoelige helpetekst beschikbaar te zijn. Hierbij is een sterke voorkeur voor de Nederlandse taal.	S
4.2.45	Voor veel voorkomende functies en voor functies die veel handelingen vragen dienen toetscombinaties (sneltoetsen) gedefinieerd te zijn. Deze dienen eenvoudig te zijn aan te passen, toe te voegen en te verwijderen.	S
Beschikbaar stellen exports (4.3)		
4.3.1	Meetdatasets, rapportages en views kunnen in diverse formaten vanuit BBMS verkregen worden middels exports. De specificatie per meetdataset benoemt deze.	M
4.3.2	Bij een export hebben gebruikers de mogelijkheid om data op de volgende wijze te selecteren: <ul style="list-style-type: none"> • Selecteren van meetdataset, (samengestelde) rapportages of views • Een (groep) van Geocodes en/of contract gebieden • Spoornummers • Kilometerlint en kilometrering • Contractgebied • Corridor • Combinaties van bovengenoemde criteria 	S
4.3.3	Per meetdataset wordt een set van standaard exports binnen BBMS beschikbaar gesteld.	S
4.3.4	De eindgebruiker heeft de mogelijkheid om zelf (ad hoc) exports op te stellen binnen een meetdataset (enkelvoudige exports).	S
4.3.5	De eindgebruiker heeft de mogelijkheid om zelf (ad hoc) exports op te stellen waarbij gegevens vanuit meerdere datasets betrokken zijn (meervoudige exports).	S

5.2.1.1 Overzicht non-functional requirements

Algemene eisen

9.1.1	Eisen uit Acceptatiecriteria KA+AB	M
9.1.2	Eisen uit Aansluitvoorwaarden ProRail	M
9.1.3	eisen uit Technology Roadmap	S
9.1.4	eisen uit Handboek Informatiebeveiliging	M
9.1.5	eisen uit Informatiebeveiligingsbeleid	M
9.1.6	eisen uit ProRail Codeerstandaarden	S

Beheer eisen

9.2.1	De applicatie moet om kunnen gaan met het mechanisme van Single Sign on (SSO)	M
9.2.2	Authenticatie (waar nodig) geschiedt via de Active Directory via LDAP protocol	M
9.2.3	Beheer moet verschillende gebruikersgroepen verschillende rechten kunnen geven voor de applicatie via een beheertools/applicatie	M
9.2.4	De applicatie wordt geïnstalleerd op een centrale server en draait op een webbrowser (thin client) op de werkplek. Schermoutput wordt gegenereerd op de centrale server. De toegepaste middleware op deze server is zo min mogelijk platformafhankelijk	M
9.2.5	Alle applicaties zijn volledig webbased of dienen binnen het project webbased gemaakt te worden. Dit geldt voor alle functies.	M
9.2.6	Active-X componenten zijn niet toegestaan.	M

9.2.7	De applicatie moet vanuit VM Ware omgeving kunnen draaien	M
9.2.8	Gebruik van de verschillende data en applicatieonderdelen moeten op gebruik en performance gemeten kunnen worden.	M
9.2.9	Performance moet SMART gemeten kunnen worden.	M
9.2.10	De applicatie heeft ruime mogelijkheden voor zowel functioneel als technische beheerders om de applicatie aan te passen aan de behoeften van ProRail. Er zijn onder meer parameters bekend en beschikbaar om de performance te kunnen optimaliseren	S
9.2.11	Als er sprake is van maatwerk wordt de sourcecode eigendom van ProRail. ProRail heeft het recht en de mogelijkheid om maatwerk functies aan te passen.	M
9.2.12	Voor printen/plotten/Exporteren wordt gebruik gemaakt van een template die de beheerder middels een GUI kan aanpassen.	M
9.2.13	Downloads door gebruikers van data uit de applicatie worden nooit op de centrale server opgeslagen maar altijd lokaal op de cliënt machine.	M

SLA eisen

9.3.1	Een standaard Service Level Agreement is beschikbaar.	S
9.3.2	Leverancier is bereid om in samenspraak met ProRail te komen tot een ProRail specifieke SLA.	M
9.3.3	Leverancier heeft aantoonbare ervaring met het invullen van de dienstverlening op basis van de standaard SLA.	M
9.3.4	Voor het oplossen van incidenten beschikt leverancier over een service organisatie die in staat is om binnen de afgesproken tijd en indien noodzakelijk ter plekke storingen aan het systeem te verhelpen.	M
9.3.5	De service organisatie beschikt over een eenduidig aanspreekpunt.	M
9.3.6	Een beschrijving van het release management proces is aanwezig.	M
9.3.7	Een beschrijving van het change management proces is aanwezig	M
9.3.8	Een beschrijving van het test proces is aanwezig en bij oplevering worden testrapporten overlegd.	M
9.3.9	De service organisatie heeft kennis geborgd en het oplossen van incidenten en uitvoering van het release proces is onafhankelijk van individuele medewerkers.	M
9.3.10	De service organisatie rapporteert aan ProRail over de overeengekomen service levels, deze rapportage bestaat in ieder geval uit een overzicht van geplande en gerealiseerde uren per melding en de status van per melding.	M

9.4.1	Voorafgaand aan opdrachtverstrekking zal ProRail de onderhoudbaarheid van het aangeboden pakket laten beoordelen door een onafhankelijke derde partij. Deze partij zal daartoe een directe analyse uitvoeren van de broncode. In bijlage Error! Reference source not found. wordt deze analyse nader toegelicht.	M
9.4.2	Met de leverancier zal een Non Disclosure Agreement worden overeengekomen dat geldig blijft zolang ProRail geen eigenaar van de software is.	M
9.4.3	De bereidheid deze beoordeling te laten uitvoeren is een "knock-out" (NoGo) criterium, hetgeen is opgenomen in de gunningscriteria.	M
9.4.4	Tijdens aanpassing van het pakket dient de leverancier bereid te zijn de kwaliteit wekelijks te laten beoordelen. Op het moment van beschikbaar stellen dient de baseline van geleverde code niet verder dan drie werkdagen in het verleden te liggen.	S
9.4.5	De leverancier dient de ProRail JAVA codeerstandaard te volgen.	S
9.4.6	Ook voor andere talen geldt dat aan deze JAVA standaards voldaan dient te worden. Tenzij deze niet generiek zijn en ook niet eenduidig vertaald kunnen worden naar toepassing binnen die andere taal.	S

Installatie eisen

9.5.1	Het systeem dient flexibel configureerbaar te zijn. Er mogen geen harde verwijzingen zijn naar driveletters, lokale printers, paden en temp bestanden.	M
9.5.2	Het systeem dient voorzien te zijn van een installatieprogramma of script waarmee alle installatiehandelingen geautomatiseerd zijn uit te voeren.	M
9.5.3	Het systeem dient voorzien te zijn van systeemdokumentatie (release documentatie, installatiedocumentatie, gebruikersdocumentatie) waarmee technisch en functioneel beheer het systeem kunnen installeren en configureren.	M

Beveiligingseisen

9.6.1	Het systeem dient aan alle eisen zoals gedefinieerd in de ProRail beveiligingsstandaard en beveiligingsbeleid te voldoen [5].	M
9.6.2	Het systeem dient de bij ProRail gebruikte Active Directory server te gebruiken voor het authenticeren en autoriseren van gebruikers.	M
9.6.3	Het systeem dient het verrichten van handelingen door gebruikers selectief toe te staan, op basis van de rechten die hen binnen een autorisatiegroep zijn toegekend. Deze rechten worden op basis van rollen en gebruikersgroepen toegekend.	M
9.6.4	Het systeem dient de toegang tot meet datasets door gebruikers selectief toe te staan, op basis van de rechten die hen binnen een autorisatiegroep zijn toegekend. Deze rechten worden op basis van rollen en gebruikersgroepen toegekend. Selectie van meetdatasets moet plaats kunnen vinden op diverse aspecten zoals: geografie, periode in de tijd, leverancier van datasets en de zoekcriteria genoemd in eis 4.2.8.	M
9.6.5	Het systeem biedt beheerders de mogelijkheid om meerdere autorisatiegroepen vast te leggen.	M
9.6.6	Het systeem biedt de mogelijkheid om meerdere gebruikers in een autorisatiegroep vast te leggen. De beheerder moet de autorisatie van een enkele gebruiker voor een nader te bepalen periode tijdelijk uit kunnen breiden onafhankelijk van de overige leden van de autorisatiegroep.	M
9.6.7	BBMS maakt voor de autorisatie gebruik van autorisatie profielen. Per autorisatie profiel kan bepaald worden welke functionaliteit ter beschikking wordt gesteld. Autorisatie profielen kunnen worden aangepast. Minimaal worden onderstaande autorisatie profielen voorzien: <ul style="list-style-type: none"> - Profiel per doelgroep benoemd in hoofdstuk Error! Reference source not found. Voor iedere leverancier wordt een afzonderlijke profiel gedefinieerd. - Applicatie beheerder & Functioneel beheerder 	S
9.6.8	Het systeem logt alle activiteiten van alle gebruikers. NB Ook de leveranciers van meetdatasets en de beheerders zijn gebruikers. Deze logging wordt ingedeeld in categorieën zodat gebruikers en beheerders snel inzicht kunnen krijgen in bepaalde type activiteiten. De logging dient zodanig te worden ingericht dat alle uitgevoerde processtappen achteraf zijn te traceren. Dit geldt ook voor de resultaten van de betreffende acties (rapportages, meldingen, etc.). De logging wordt weergegeven als leesbare tekst (niet gecodeerd).	M
9.6.9	Toegang tot het systeem van buiten ProRail is alleen mogelijk met behulp van een token. De Rechten voor externe (buiten ProRail) gebruikers moeten volgens de standaard werkwijze worden toegewezen (RSAKey).	M

Performance eisen

9.7.1	Bij het opleveren van een release naar de acceptatieomgeving dient een performance testrapportage opgeleverd te worden met daarin resultaten van de latency- load- en stresstesting.	M
9.7.2	De applicatie dient schaalbaar te zijn, door het inzetten van meerdere servers. Daarbij moet load balancing kunnen worden toegepast.	M
9.7.3	Wanneer grote meetdatasets ingelezen worden mogen de responsetijden voor andere gebruikers stijgen met maximaal 20%. Dit geldt ook video en fotobestanden.	S
9.7.4	Om performancerisico's te voorkomen dienen video en fotobestanden automatisch (zonder interactie van beheerders) te kunnen worden ingelezen buiten kantooruren.	M
9.7.5	Uit performanceoverwegingen worden Videobestanden alleen in gecomprimeerde vorm ontsloten. Transcodering vindt plaats mbv een veelgebruikte codec die deel uitmaakt van de gewenste Windows versie uit de Aansluitvoorwaarden ProRail [2] en op zodanige wijze dat gebruikers geen performance effect bemerken. Om aan deze eis tegemoet te komen kan transcodering op een andere server, of buiten kantooruren worden uitgevoerd.	S
9.7.6	Als tijdens het synchroon tonen van baanbeelden, diagrammen en geografische beelden de selectie in een viewer verandert dienen de drie genoemde presentatievormen in 90% van de gevallen binnen 3 seconden ververs te worden en synchroon te blijven lopen.	S
9.7.7	Als de selectie in één viewer verandert dienen de andere viewers in 90% van de gevallen binnen 5 seconden gesynchroniseerd te worden.	S
9.7.8	Het systeem is voorzien van een wachttijd indicatie als de wachttijd langer is dan de hierboven genoemde responsetijden.	S
9.7.9	Bovengenoemde responsetijden gelden ook indien 200 gebruikers tegelijk het systeem gebruiken en van die 200 gebruikers maximaal 40 gebruikers tegelijkertijd video- of fotobestanden bekijken.	S

Hulp- en opleiding eisen

9.8.1	Het systeem dient zodanig opgezet te worden dat bijvoorkeur geen, maar maximaal een halve dag opleiding nodig is voor de gebruikers.	S
9.8.2	Het systeem dient opgeleverd te worden met een online help functie. De online help functie dient in het Nederlands te zijn. Bronteksten van de online help dienen meegeleverd te worden zodat de beheerder deze kan aanpassen.	M
9.8.3	Een quick reference card met een opsomming en beknopte beschrijving van de belangrijkste functies dient opgeleverd te worden.	M
9.8.4	Handleiding voor gebruikers dient in documentvorm te zijn. Handleiding bevat werking van de verschillende functies en instructies hoe en onder welke voorwaarden die functies moeten en kunnen worden uitgevoerd.	M
9.8.5	Brontekst van de handleiding dient meegeleverd te worden zodat de beheerder deze kan aanpassen, bijvoorbeeld in Word formaat.	M
9.8.6	Het helpsysteem dient bij raadpleging automatisch de relevante informatie te selecteren afhankelijk van de getoonde informatie („context sensitive help“).	S
9.8.7	Bij opleveren van nieuwe releases in de acceptatieomgeving is geen opleiding nodig.	S
9.8.8	Tegelijkertijd met de release worden releasenotes uitgebracht voor beheerders en gebruiker en worden alle hulp en opleidingfuncties en alle documentatie geactualiseerd.	M
9.8.9	De naamgeving binnen BBMS komt overeen met de binnen ProRail gangbare naamgeving voor objecten, parameters binnen een dataset, rapportages en views.	S
9.8.10	De bediening in BBMS komt overeen met de bediening van standaard MS Windows applicaties. Waar mogelijk wordt grafisch gewerkt of grafisch ondersteund.	S
9.8.11	Foutmeldingen zijn in het Nederlands en geven volledige situatiespecifieke informatie over de oorzaak, locatie en moment van optreden van de fout. De foutmeldingen worden weergegeven in leesbare tekst (niet gecodeerd).	M

Ontwikkelmethodiek

9.9.1	Alle nieuwe of aangepaste functionaliteit zal na vrijgave onderdeel uitmaken van het standaardpakket.	M
9.9.2	Ontwikkeling dient op een transparante en traceerbare manier plaats te vinden. Dit betreft onder andere het vastleggen van besluiten en ontwerpkeuzes, het tijdig opleveren van documentatie en het proactief communiceren over resultaten en afwijkingen van de planning.	M
9.9.3	Er dient voldoende gelegenheid te zijn voor gebruikerstesten. Daarbij gaan we uit van een iteratief ontwikkelproces met iteraties van 2 of 3 weken en per iteratie 2 resp. 3 dagen gelegenheid tot testen.	M
9.9.4	De leverancier dient te beschrijven welk deel van het testproces zij voor haar rekening neemt en wat zij van ProRail verwacht.	M
9.9.5	Tijdens ontwikkeling wordt alle code bij voorkeur dagelijks gecompileerd en gebouwd. Wekelijks wordt de code aangeboden ter beoordeling van de codekwaliteit en het voldoen aan afspraken ten aanzien van architectuur en codeerstandaards (zie Onderhoudbaarheid).	S
9.9.6	De leverancier dient het door hem gehanteerde werkwijze en haar ontwikkelproces beschreven te hebben en aan ProRail ter beschikking te stellen.	M
9.9.7	Onderwerpen die in de werkwijzebeschrijving aan bod dienen te komen zijn: Projectaanpak, Risicobeheersing, Configuratiebeheer, Kwaliteitsbeheersing (waaronder reviewmethode), Communicatiemethode, Projectbeheersing en beheer van eisen, wijzigingen en ontwerpkeuzes.	M
9.9.8	Per processtap moet duidelijk beschreven zijn wat start- en afsluitcondities zijn en welke producten opgeleverd worden. Hierbij gaat het om producten in ruime zin zoals software, documenten, documentatie, risicoregisters, kwaliteitsregisters en configuratieoverzichten. Uit de beschrijving dient expliciet naar voren te komen wanneer er oplevermomenten zijn en wanneer van ProRail input of besluitvorming gevraagd wordt.	M
9.9.9	De leverancier stemt de werkwijze bij implementatie af met ProRail.	M

BBMS Functioneel Ontwerp

Inleiding

In workshop op 5 juli is het thema dekkingslagen besproken. In deze workshop kwam naar voren dat in BBMS in eerste instantie niet wordt gepland. De benodigde data voor dekkingslagen, zoals contract en planning worden aangeleverd.

Als leidraad voor de structuur van deze notitie zijn de eisen van het PvE genomen. Doel van deze notitie is een beeld te geven hoe dekkingslagen kan worden ingericht in BBMS. Het bijdragen aan de beeldvorming voor toekomstige invulling.

Eisen PvE

Voor dekkingslagen zijn onderstaande eisen opgenomen in het PvE BBMS versie 2.0.

2.1.2	Om inzicht te houden in de te verwachten meetdata en de realisatie van de gecontracteerde meetdiensten wordt op de asset data in verschillende lagen de dekking aangegeven voor de op te nemen meetdata. Deze lagen zijn: Contract Planning Uitvoering Oplevering meetgegevens Dekking aanbrengen houdt in dit verband in het per meetdataset of parameter van een meetdataset aan een laag toewijzen van delen van de railinfrastructuur. BBMS is ingericht om op de asset data in 4 lagen dekking aan te kunnen brengen.
2.1.3	Daar het contract kan bestaan uit meerdere metingen per jaar, wordt op de dekkingslaag contract de frequentie aangegeven. Voor de dekkingslagen planning, uitvoering en meetgegevens zijn per laag meerdere dekkingen per jaar beschikbaar.
2.1.4	Dekkinglagen kunnen zowel handmatig aangebracht worden in BBMS zelf als geïmporteerd uit een externe bron.
2.1.5	Het handmatig vullen van een laag wordt gedaan op grafische kaarten door het aanklikken van de delen van de railinfrastructuur (per USIS-id of alle USIS-id's binnen een door de gebruiker aan te duiden bereik), waarmee dat element toegewezen wordt aan een van de dekkingslagen.
2.1.6	Dekkinglagen kunnen eenvoudig toegewezen worden aan individuele parameters, aan een meetdataset of aan meerdere meetdatasets tegelijk.
2.1.7	Gebruikers kunnen vergelijkingen maken tussen verschillende lagen waarbij rekening gehouden wordt met de frequentie per jaar en de meerdere dekkingen per laag. Hiermee is BBMS in staat te beoordelen of bijvoorbeeld een spoor met frequentie 3x per jaar in de contractlaag afgedekt wordt door 3 dekkingen in planning, uitvoering of meetgegevens.
2.1.8	De verschillen in dekking worden zowel tabellarisch als grafisch weergegeven.
2.1.9	De dekkingslagen kennen binnen BBMS gelijke beheers- en gebruiksfuncties als de meet- en metadata, zoals autorisaties, instellen tijdsperiode, exports etc.
2.2.1	Bij het innemen van meetdata wordt bepaald in hoeverre de afspraken uit het meetcontract al zijn nagekomen. De planning van de te meten baanvakken en sporen wordt daarbij vergeleken met de sporen waarvan meetgegevens zijn ingenomen. Voor sporen waar "gaten" in de meetdata zijn ontstaan wordt de dekkingsgraad aangegeven (het aantal en omvang van de "gaten")

Uitgangspunten/aanname

Tijdens de workshop zijn de volgende aannames gemaakt:

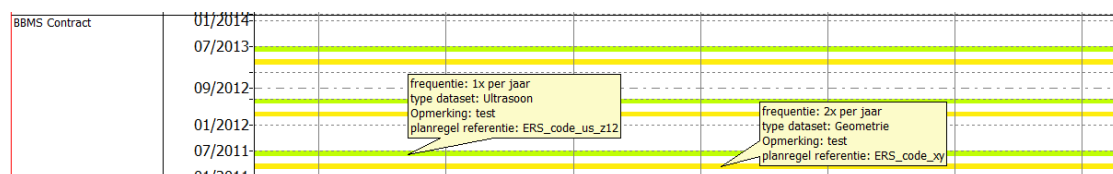
- De data voor dekkingslagen wordt aangeleverd door derden. Wijze van aanlevering en verwerking daarvan staat nog open. Zal in de toekomst door middel van datasetbeschrijving worden uitgewerkt.
- In BBMS vindt geen planningsoptimalisatie plaats (wordt niet als planningstool ingezet)
- Het accent ligt vooral op het visualiseren.

Onder een dekkingslaag wordt verstaan een gegevenstype, die 'iets' aantoont voor een bepaald gebied (lees USIS element), met een bepaald tijdstip

Voorbeeld invulling

2.1.3	Daar het contract kan bestaan uit meerdere metingen per jaar, wordt op de dekkingslaag contract de frequentie aangegeven. Voor de dekkingslagen planning, uitvoering en meetgegevens zijn per laag meerdere dekkingen per jaar beschikbaar.
-------	---

In BBMS is het mogelijk om in een diagram het contract weer te geven, waarbij de kleur de frequentie aangeeft, zoals in het onderstaande voorbeeld is aangegeven:



Figuur 1: voorbeeld diagram weergave dekkingslagen met verschillende frequentie

Grenswaarde niveau	Voorwaarde
<input checked="" type="checkbox"/> 2x per jaar	TH = 2x per jaar
<input checked="" type="checkbox"/> 4x per jaar	TH = 4x per jaar
<input checked="" type="checkbox"/> 1x per jaar	TH = 1x per jaar
<input checked="" type="checkbox"/> Overige	TH = overige

Figuur 2: legende kleurbetekenis

Via het navigeren met de muis op betreffende diagram, kan aanvullende informatie worden getoond.

2.1.4	Dekkinglagen kunnen zowel handmatig aangebracht worden in BBMS zelf als geïmporteerd uit een externe bron.
-------	--

In principe kunnen we dekkingslagen data beschouwen als een soort van meetdata. Daarmee is het ook mogelijk deze data te importeren, zoals beschreven in het FO document **02 Concept-Automated import - Geometry+Pictures_v10 ProRail reviewed**.

Het ligt voor de hand dat voor deze data dan ook dataset beschrijving aanwezig moet zijn.

Naast de geïmporteerde dekkingslagendata kan deze data ook gewijzigd worden in de standaard gegevenseditor.

BBMS contract dekking										
Record	trajectnaam	spoor	Gebeurtenis (Datum)	Gebeurtenis (Tijd)	Van	Tot	frequentie	type dataset	Opmerking	planregel referentie
1	013	A	3-3-2013	11:25:25	77.700	99.900	2x per jaar	Geometrie	test	ERS_code_xy
2	013	A	3-3-2011	11:25:25	77.700	99.900	2x per jaar	Geometrie	test	ERS_code_xy
3	013	A	3-3-2012	11:25:25	77.700	99.900	2x per jaar	Geometrie	test	ERS_code_xy
4	013	A	5-6-2011	11:25:25	77.700	99.900	1x per jaar	Ultrasoon	test	ERS_code_us_z12
5	013	A	5-6-2013	11:25:25	77.700	99.900	1x per jaar	Ultrasoon	test	ERS_code_us_z12
6	013	A	5-6-2012	11:25:25	77.700	99.900	1x per jaar	Ultrasoon	test	ERS_code_us_z12

Figuur 3: Voorbeeld van gegevenseditor om data handmatig in te voeren of wijzigen.

2.1.2	<p>Om inzicht te houden in de te verwachten meetdata en de realisatie van de gecontracteerde meetdiensten wordt op de asset data in verschillende lagen de dekking aangegeven voor de op te nemen meetdata. Deze lagen zijn:</p> <ul style="list-style-type: none"> Contract Planning Uitvoering Oplevering meetgegevens <p>Dekking aanbrengen houdt in dit verband in het per meetdataset of parameter van een meetdataset aan een laag toewijzen van delen van de railinfrastructuur. BBMS is ingericht om op de asset data in 4 lagen dekking aan te kunnen brengen</p>
-------	--

De informatie wordt per dekkingslaag opgeslagen in een gegevenstype. Hiermee is dan ook gelijk mogelijk de verschillende lagen te creëren.

Grafisch kan dit in een diagram getoond worden, waarbij op de verticale as de tijd wordt getoond en de horizontale as het betreffende traject. Daarnaast krijgt iedere laag zijn eigen kleur, om deze kunnen onderscheiden. In principe zou ook per dekkingslaag een diagram worden ingericht. Inrichting is mede afhankelijk van het doel van het diagram. Combinaties zijn ook mogelijk.

Bijlage III Requirements SBG++

Het project SBG++ heeft de Military Standard 498 naar eigen zeggen gevolgd. De inhoud van de documenten voldoet weliswaar niet helemaal aan hetgeen de data item descriptions van de Military Standard 498 voorschrijven, maar de nodige documentatie is wel opgesteld. De unieke identificatie van requirements is niet zuiver doorgevoerd. Er is wel gewerkt met een User Stories nummering maar deze is niet overal te herleiden. Onderstaand volgen voorbeelden van de verschillende documentatie die is opgesteld en geven inzicht in de wijze waarop de requirements zijn vastgelegd en hoe bijvoorbeeld de User Stories nummering is gehanteerd.

Operational Concept Description

In het OCD word een systeembeschrijving gegeven van het huidige systeem met taken en het nieuwe systeem. De verschillende rollen worden bijvoorbeeld beschreven.

Planner

Bewaakt het plan op uitvoerbaarheid en onderhoudt het door:

- *Effect VKL vertragingberichten in te schatten (rijwegconflict - wie eerst, spoorbezettingconflict, aansluitingen) en zo nodig plan daarop aan te passen*
- *Verwerken VKL dienstregelingberichten n.a.v verstoringen*
Verwerken effecten bijzondere voorvallen in het plan zoals:
- *Effecten van storing in de infra op de dienstregeling onderkennen en verwerken*
- *Effecten van storing in de infra op het rijwegplan onderkennen en verwerken*

Maar ook de systeemondersteuning van het huidige systeem wordt per taak omschreven.

Bewaken uitvoerbaarheid plan door planner lange termijn (C)

Planner lange termijn bewaakt binnen zijn tijdscope of het plan uitvoerbaar is/blijft. Hij doet dit door de planregels in de planvensters te toetsen op uitvoerbaarheid naar aanleiding van verstoringen in de treindienst die hij veelal verneemt van de verkeersleider of verstoringen binnen zijn gebied die hij afstemt met zijn omgeving.

Er wordt een rechtvaardiging uiteen gezet voor het te bouwen systeem. Bijvoorbeeld:

2. Knelpunten in systeem functionaliteiten [1] Het systeem biedt geen optimale / niet voldoende ondersteuning bij de taken die de treindienstleider moet uitvoeren.

Door middel van een tabel worden de verschillende knelpunten met het huidige systeem weergegeven en wordt aangegeven wat wel/niet wordt opgelost.

SBG01	De juistheid en volledigheid van het geleverde procesplan wordt onvoldoende toetsbaar gepresenteerd in de planvensters en SBG.	Door het procesplan van een PPLG volledig en in samenhang op een grafische wijze te presenteren is voor elke activiteit duidelijk zichtbaar of de planregels volledig en juist zijn.
-------	--	--

Met behulp van printscreens worden vervolgens een groot aantal scenario's geschetst en complexe praktijkvoorbeelden uitgediept. Dit alles tekstueel in het OCD.

Software Requirements Specification

In het SRS worden de User Stories beschreven, onderstaand twee voorbeelden.

Sporen

SBG++ toont een grafiek met daarin alle relevante sporen naast elkaar. Deze sporen worden weergegeven als verticale lijnen. In deze grafiek loopt de tijd van beneden naar boven (US039).

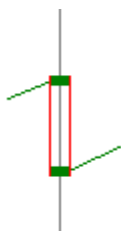
Langs de bovenrand van de grafiek worden de spoornamen getoond, langs de linkerrand de tijdstippen. Rechts en onder zit een scrollbar.

In de oude SBG loopt de tijd van links naar rechts, en staan de sporen boven elkaar. De belangrijkste reden om bij SBG++ de sporen langs de x-as te tonen is dat er op die manier meer sporen kunnen worden getoond (althans bij een landscape beeldscherm). Meer sporen kunnen tonen is nodig omdat SBG++ niet alleen perronsporen laat zien, maar ook alle overige sporen. (Daarnaast geldt dat het aantal sporen in de oude SBG sowieso al onvoldoende was.)

De default afstand tussen sporen is 4 staafbreedtes. (Wat een staafje is wordt uitgelegd in het hoofdstuk over bezettingbeelden.) Deze afstand is configureerbaar. De gebruikte eenheid daarbij is de breedte van een staafje. Dat zorgt ervoor dat de afstand tussen sporen automatisch meeschaalt als de staafjes breder of smaller worden gemaakt.

Aankomst na vertrek

In de planning kan het voorkomen dat een aankomst zozeer vertraagd is dat de aankomsttijd na de vertrektijd komt. In dat geval wordt het staafje rood gekleurd (US343) (US966).



Deze situatie wordt niet getekend als een bezetting omdat er feitelijk geen bezetting is gedurende die tijd: een trein kan niet een spoor bezetten voorafgaand aan de aankomst. Als tussen aankomst en vertrek een andere activiteit (b.v. een doorkomst) plaatsvindt op hetzelfde spoor, dan wordt dit niet beschouwd als een conflict.

Er kan een mix ontstaan van rode en groene staafjes in een bezettingbeeld. Neem als voorbeeld de onderstaande splitsing: trein 12 wordt gesplitst in treinen 14 en 16:

Aankomst	12	12:30
Vertrek	14	12:34
Vertrek	16	12:40

Als alles volgens plan verloopt is dit het bezettingbeeld:



(De tijdstippen dienen ter verduidelijking. In SBG++ worden deze niet getoond.)

Echter, trein 12 heeft 7 minuten vertraging:

Aankomst	12	12:37
Vertrek	14	12:34
Vertrek	16	12:40

Dit wordt als volgt getoond:



Het linker staafje symboliseert nog steeds de materieelrelatie tussen trein 12 en trein 14, maar omdat de aankomst na het vertrek volgt (volgens het plan, in het echt kan dat natuurlijk niet) wordt het niet als bezetting getoond.

Het rechter staafje symboliseert de materieelrelatie tussen trein 12 en trein 16. Dit is een normaal staafje: aankomst voor vertrek, dus niet ondersteboven en groen gekleurd.

Zoals te zien is zijn de User Stories tekstueel, detaillistisch van aard en waar nodig geacht voorzien van een afbeelding of scherm prints.

Interface Requirements Specification

Er is een IRS opgesteld waarin alle eisen gerelateerd aan de interface met de Enterprise Service Bus en SBG++ systeem componenten worden beschreven. Deze eisen zijn tekstueel.

Enkele voorbeelden van deze eisen:

Cardinaliteit

Voor alle SBG++ instanties op een VL post zal 1 Consuming Endpoint (CEP) worden uitgevoerd. Dit endpoint zal uit beschikbaarheidsoogpunt redundant worden uitgevoerd. De implementatie van het endpoint dient voorbereid te zijn op horizontaal schalen.

I-BESCH-1 Het SBG++ Consuming endpoint dient redundant uitgevoerd te worden. Een evt. falen van het endpoint mag voor een consumer geen impact hebben.

I-CARDI-1 De implementatie van het SBG++ consuming endpoint (CEP) dient voor load balancing doeleinden voorbereid te zijn voor horizontaal schalen. Dit dient transparant te zijn voor de implementatie van SBG++.

I-CARDI-2 Op elke VL post zal één (redundant) SBG++ Consuming Endpoint uitgevoerd worden voor de op die post aanwezige SBG++ instanties.

Het komt er in feite op neer dat er per EMS regio een SBG++ endpoint wordt uitgevoerd. Voor een evt. SBG++ applicatie op de centrale Post21 locatie kan evt. ook een endpoint worden uitgevoerd.

Vereiste Toestanden

I-INTSTART-1 Bij het opstarten van een SBG++ instantie of het CEP dient ten allen tijde de interface weer volledig operationeel te worden, onafhankelijk van de volgorde van opstart.

Voor zowel CEP als afnemers mogen geen aannames worden gedaan over het al dan niet aanwezig zijn van de ander. Willekeurig welke partij het eerst gestart wordt, uiteindelijk dienen de interfaces automatisch actief te worden.

Opmaak Berichten

I-BERICHT-1 De request-reply berichten (de http -) op de interface dienen te voldoen aan ProRailML richtlijn. Voor deze interface gaat het om de volgende berichten:

SubscribeRequestARP

SubscribeResponseARP
RenewARP
RenewResponseARP
UnsubscribeRequestARP
UnsubscribeResponseARP
GetTopicsRequestARP
GetTopicsResponseARP

Voldoen aan de ProRailML richtlijn betekent concreet dat berichten SOAP berichten moeten zijn, de SOAP header WSA-Addressing velden bevat en dat de inhoud van de SOAP body voldoet aan het Canonical Data Model (CDM).

Voor asynchrone berichten (Notify) wordt hiervan afgeweken. De inhoud van de EMS body moet wel aan het CDM voldoen.

De request-reply berichten zijn vastgelegd in de WSDL in Bijlage A.

I-BERICHT-2 De request-reply berichten op de interface dienen te voldoen aan de SOAP 1.2 standaard.

*I-BERICHT-3 In de SOAP header van de berichten dient gebruik te worden gemaakt van WS-Addressing berichtvelden volgens de specificatie **Error! Reference source not found.***

De SOAP binding voor WS-Addressing is nodig om een transport onafhankelijk routing te kunnen realiseren op de ESB. Deze eis is ook onderdeel van de ProRailML standaard.

Bijlage IV Requirements Routelint

Operational Concept Description

In het project Routelint is een Operational Concept Descriptions opgesteld. Zoals met de volgende printscreens wordt aangetoond is zo'n Operational Concept Description geheel volgens de standaard opgesteld en is eventuele afwijking expliciet vermeld.

OCD-DID.PDF (BEVEILIGD) - Adobe Reader

Bestand Bewerken Beeld Document Opties Venster Help

1 / 7 96.5% Zoeken

DATA ITEM DESCRIPTION

Form Approved
OMB NO 0704-0188

Public reporting burden for collection of this information is estimated to average 110 hours per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. TITLE
OPERATIONAL CONCEPT DESCRIPTION (OCD)

2. IDENTIFICATION NUMBER
DI-IPSC-81430

3. DESCRIPTION/PURPOSE

3.1 The Operational Concept Description (OCD) describes a proposed system in terms of the user needs it will fulfill, its relationship to existing systems or procedures, and the ways it will be used.

3.2 The OCD is used to obtain consensus among the acquirer, developer, support, and user agencies on the operational concept of a proposed system. Depending on its use, an OCD may focus on communicating the user's needs to the developer or the developer's ideas to the user and other interested parties. The term "system" may be interpreted to apply to a portion of a system.

4. APPROVAL DATE
(YYYYMM) 941205

5. OFFICE OF PRIMARY RESPONSIBILITY
EC

6a. DTIC APPLICABLE

6b. GIDEP APPLICABLE

Bladwijzers

- Cover Page (front)
- Cover Page (back)
- 1. Scope
 - 1.1 Identification
 - 1.2 System overview
 - 1.3 Document overview
- 2. Referenced documents
- 3. Current system or situation
- 4. Justification for and nature of changes
- 5. Concept for a new or modified system

Figuur 65 "Voorbeeld Data Item Description Operational Concept Description"

P21HSM-OCD-030.pdf - Adobe Reader

Bestand Bewerken Beeld Document Opties Venster Help

6 / 72 143% Zoeken

1.3 Documentoverzicht

Een OCD wordt gebruikt om overeenstemming te bereiken tussen opdrachtgever, opdrachtnemer, gebruikersorganisatie en beheerorganisatie. Het document geeft achtereenvolgens een:

- Beschrijving van de huidige situatie (hoofdstuk 3)
- Rechtvaardiging en aard van de wijzigingen (hoofdstuk 4)
- Beschrijving van het operationele concept voor het nieuwe systeem (hoofdstuk 5)
- Beschrijving van operationele scenario's (hoofdstuk 6)
- Samenvatting van de gevolgen van het nieuwe systeem (hoofdstuk 7)
- Analyse van het nieuwe systeem (hoofdstuk 8)

1.4 Afwijkingen van de standaard

Dit document is opgesteld conform DOD standaard MIL-STD-498, met betrekking tot DID nummer DI-IPSC-81430 voor een OCD. In afwijking op deze standaard zijn paragrafen 4.2 en 5.3 aangevuld met mogelijke uitbreidingen die na fase 1 (Rijden met RouteLint) zouden kunnen worden opgenomen.

Figuur 66 "Operational Concept Description Routelint"

System/Subsystem Specification

De System/Subsystem Specification bevat alle eisen in uitgeschreven vorm. Elke eis heeft een prefix gekregen. Zoals in het voorbeeld de eerste algemene eis, hier is de prefix ALG gekozen.

3.2 Functionele eisen

3.2.1 Algemeen

In deze paragraaf worden alle aan OOGST gestelde functionele eisen genoemd die voor meer dan één afnemer / concept gelden.

ALG.1

Ontsluiting van Post21

a. OOGST dient functionaliteit te bevatten voor het generiek ontsluiten van Post21.

Op het moment dienen met name de volgende systemen ontsloten te worden.

1. Procesleiding Rijkswegen (plangegevens)
2. Het treinvolgstelsel (treinposities)
3. Treinbeheersingsystemen (actuele infratoestand)

De volgende systemen dienen gelijktijdig ontsloten te kunnen worden.

- a. EBP
- b. KEV
- c. KBV

b. OOGST dient tevens de functionaliteit van IVA Post als leverancier van postgeoriënteerde gegevens te bevatten. Dit betreft dus de volgende zaken:

1. Afleidingen.
2. Interfaces.

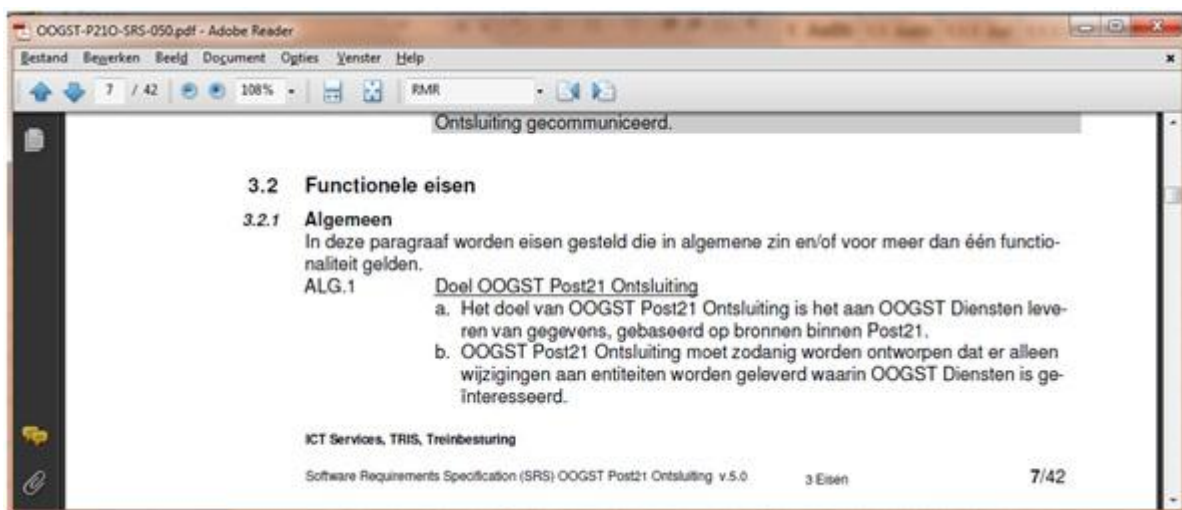
Toelichting:

Het is de bedoeling dat OOGST functionaliteit bevat waarmee het kan dienen als enkel en in principe generiek systeem voor ontsluiting van gegevens uit het traditionele, postgeoriënteerde Post21 (d.w.z. in Post21-specifieke taal en niet dienstgeoriënteerde) en met name Post21 Treindienstleiding: PRL, het treinvolgstelsel (TNV of TROTS) en de TBS-systemen (EBP, KEV en KBV; t.z.t. ook ASTRIS). Gezien de lopende migratiestrategie van TNV naar TROTS is het niet zinvol te eisen dat zowel met TNV als met TROTS kan worden omgegaan. Dat kan op dit niveau ook in het midden worden gelaten, maar als op een lager niveau eisen TNV- of TROTS-specifiek moeten worden, dan betekent het dat er verschillende versies moeten zijn van die eisende documenten (en de bijbehorende ontwerpdocumenten en software): in eerste instantie voor het gebruik TNV en daarna een gewijzigde versie voor het gebruik van TROTS. Er is niet gekozen voor het in de eisende documenten in de eisen onderscheiden tussen de situatie van het gebruik van TNV en de situatie van het gebruik van TROTS omdat er dan ook verschillende versies zullen zijn van ontwerpdocumenten en software, maar er na het uitschakelen van TNV "dode" informatie zou achterblijven in de eisende documenten.

Figuur 67 'SSS Routelint voorbeeld Prefix'

Software Requirements Specification

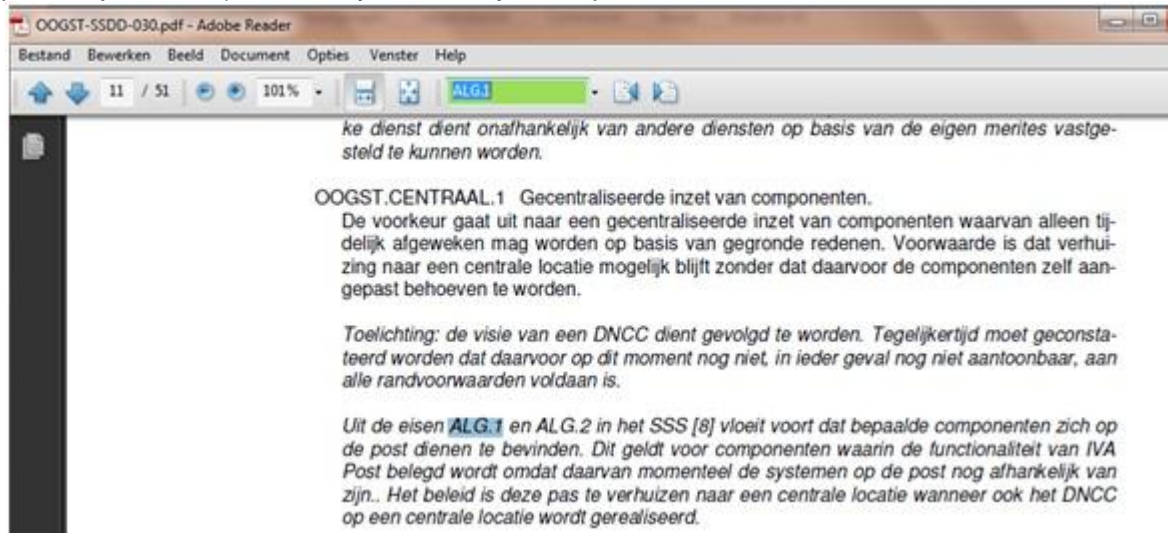
In de Software Requirements Specification zijn de eisen op vergelijkbare wijze uitgeschreven als in de System/Subsystem Specification. Het document is ook hier volledig opgesteld volgens de Data Item Description. Ook de prefix die gehanteerd is, is gelijk.



Figuur 68 'SRS Routelint'

System/Subsystem Design Description

In de System/Subsystem Design Description wordt eveneens gebruik gemaakt van de dezelfde prefix als in de specificatie documentatie. Bij ontwerp keuzes of architectuur ontwerpen wordt (via deze prefix bijvoorbeeld) naar de System/Subsystem Specification verwezen.



Figuur 69 "Ontwerpkeuze met referentie naar SSS"



Figuur 70 "Architectuur ontwerp met verwijzing naar SSS"

Bijlage V Requirements RRCB

Om inzicht te geven in vastlegging van requirements bij het project RRCB volgen nu enkele voorbeelden.

High level business requirements zijn vastgelegd in tabellen waarbij wordt aangegeven wat de requirement is, wie de stakeholder is, of de huidige applicatie voorziet en wat het gevolg zal zijn. De requirements zijn voorzien van een nummer.

High level requirements	Stakeholder	RRCB dekt functionaliteit	Gevolg
1 Vastleggen PGO contractafspraken	Controller	<i>Mix maat. Het bedrag voor de totale looptijd moet worden ingevoerd. Deze wordt verdeeld door het systeem naar de onderliggende maanden. Specifieke contractafspraken (bv verschillende startdata van overgang contractgebieden) kunnen niet worden vastgelegd</i>	<ul style="list-style-type: none"> als de geautomatiseerde verdeling lineair niet juist is dan kan dit handmatig worden aangepast. Dit is echter niet efficiënt en is foutgevoelig. systeem is star
2 Vastleggen en muteren vaste mutanten uit PGO contracten. Daarbij inzicht in de wijzigingen van vaste mutanten binnen een PGO contract gedurende contractperiode.	Controller	<i>Onvoldoende: ProRail kent verschillende vaste mutanten (maandbonus, 2jaarlijkse bonus, tijdelijke inhouding en malus) en kan deze nu niet goed vastleggen. Iedere mutant moet bedrag totaal / looptijd worden uitgerekend. RRCB verdeelt dit bedrag lineair over de totale contractduur ipv ingangsmaat. Tab mutatie betaalpost kan komen te vervallen.</i>	<ul style="list-style-type: none"> Niet inzichtelijk en rommelig, er mist een overzicht per periode met basisbedrag + mutanten Veel hand- en zoekwerk de geautomatiseerde verdeling lineair is niet juist. Dit kan handmatig worden aangepast. Dit is echter niet efficiënt en is foutgevoelig.
3 Vastleggen en muteren van tijdelijke mutanten als voorstel Prestatie Verklaring. De tracemanager accordeert of wijkt af. Achteraf moet inzichtelijk zijn wat voorstel van Controller was en wat definitief besluit Tracemanager was.	Controller	<i>Onvoldoende: Is niet mogelijk om per maand de tijdelijk mutanten aan te geven. Bv deze maand 3% bonus ipv 5%. Systeem onderscheidt niet voorstel PV en definitief PV (door Tracemanager). Het basis maandbedrag kan nu of wel of niet betaald worden. Dit moet opsplitsbaar worden in bv 75%/25%. Mutanten (bonus/malus) moeten nu opgeteld worden tot 1 bedrag. Dit moet op te splitsen zijn zodat mutanten inzichtelijk blijven.</i>	<ul style="list-style-type: none"> Met vastleggen kunstwerk de maandbedragen invoeren en bijwerken kost meer tijd dan werken zonder systeem. inzichtelijkheid achteraf is vereiste
4 Inzicht in de opstaande afwijkingen per contract	Controller	<i>Onvoldoende: Op dit moment niet mogelijk om afwijkingen te koppelen aan contractstukken (bv afwijking op veiligheid). Dit heeft namelijk invloed op verschillende bonus/malus regelingen. Totaal opstaande afwijkingen is geen informatie.</i>	<ul style="list-style-type: none"> Kost veel tijd om overzicht te krijgen. Deze moet handmatig aangepast worden. Volledigheid??

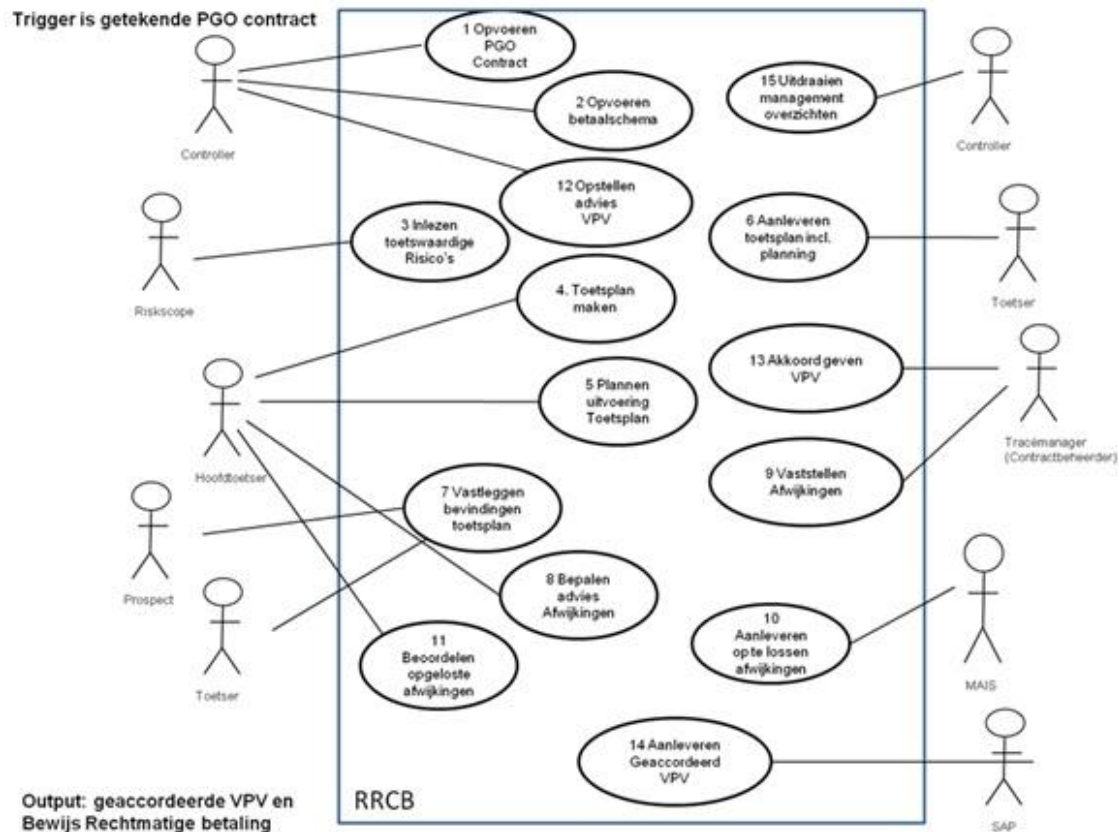
Tabel

1'High level requirements project RRCB'

Use Cases RRCB

Met de start van het project is begonnen om de High Level Requirements zoals in tabel 1 weergegeven te verzamelen. Op basis van gesprekken met gebruikers zijn ook diverse Use Cases opgesteld. Deze worden voorafgegaan door een Use Case diagram.

5. De grote rechthoek rondom de usecases geeft de scope het systeem RRCB aan.



Figuur 71 'Use Case Diagram RRCB'

UC13. Goedkeuren advies Prestatie Verklaring (processtap Prestatieverklaren)	
Actor(en)	Contractbeheerder
Doel	Definitief vaststellen van Prestatieverklaring conform contractafspraken en prestaties ON
Omschrijving	Aan de hand van contractafspraken en prestaties van voorgaande periode bepaald de controller de tijdelijke mutanten die leiden tot maandfacturen.
Trigger	Advies Verzoek Prestatie Verklaring (VPV) beschikbaar
Preconditie	VPV ontvangen Inzicht in openstaande afwijkingen per contract Inzicht in relevante contractafspraken
Postconditie	Maandbedrag kan door SAP gefactureerd worden
Requirement	13.1 Opvragen advies PV en afdrukken advies PV 13.2 Alleen contractbeheerder van betreffende contract mogen advies PV goedkeuren. 13.3 Er is een vrij tekst veld waarin Contractbeheerder zijn besluit kan toelichten 13.4 De Contractbeheerder kan afwijken van het advies dat gedaan is door risicomanager, hoofdtoetsen en controller. Dit gebeurt door wijzigen van velden die tot maandbedragen leiden (percentages, inhoudingen/n etc) 13.5 De advies PV blijft inzichtelijk en is achteraf herleidbaar. 13.6 De Contractbeheerder geeft een toelichting bij de wijzigingen op advies PV 13.7 Zodra Contractbeheerder PV goedgekeurd heeft, kan er niet meer gewijzigd worden. 13.8 De definitieve PV wordt doorgegeven aan SAP ter facturatie aan de ON.

Figuur 72 'Voorbeeld Use Case project RRCB'

System/Subsystem Specification

In het project heeft men in eerste instantie gekozen voor het toepassen van de Military Standard 498. Dit als gevolg van het feit dat het project in eerste instantie gestart is als samenwerking tussen ProRail en Rijkswaterstaat. Deze samenwerking is na vrij korte tijd beëindigd.

Als onderdeel van de keuze voor deze standaard is een Software Subsystem Specification opgesteld. In dit document zijn alle requirements opgenomen in hoofdstuk 3 zoals de Military Standard 498 voorschrijft. De hoofdstukken die aandacht geven aan kwalificatie en traceability zijn niet opgenomen in het document. De requirements zijn vastgelegd in tabelvorm zoals in de afbeelding 36 te zien is of in als Use Case beschrijving welke veelal zijn overgenomen uit het document met Use Cases.

3 Eisen

In dit hoofdstuk worden de eisen per module uitgewerkt.
Zie Tabel 1 voor de te onderscheiden modules.

Tabel 1: Toelichting op eisennummer

Indeling/Modulenaam	Use Casenummer	Voorbeeld nummering hoofdeisen	Voorbeeld nummering sub-eisen
Algemeen		0-001	0-001-1
Contract	1, 2	1-001	1-001-1
Risico's & toetsen	3, 4, 5, 6	3-001	3-001-1
Afwijkingen	7, 8, 9, 10, 11	7-001	7-001-1
Prestatieverklaring	12	12-001	12-001-1
Niet-functionele eisen	ASE	ASE-001	ASE-001-1

3.1 Algemene eisen

Eisnummer	Eis	Toelichting	Kwalificatie	MoSCoW
0-001	De applicatie opent bij opstarten altijd in het RRCB startscherm		Test	S
0-002-1	In het RRCB startscherm worden de volgende items weergegeven: - Naam van de applicatie - Versie nummer van de applicatie		Test	S
0-002-2	In het RRCB startscherm moet de mogelijkheid gegeven worden (b.v. knoppen/tabbladen) voor het benaderen van de overige modules		Test	S
0-003	Het moet mogelijk zijn om gebruikersprofielen te definiëren gebaseerd op rollen en rechten.		Test	M
0-004	Alle objecten moeten uniek te identificeren zijn.	Risico, toets, toetsplan, toetsplanning, constatering, bevinding, afwijking/tekortkoming		M
0-005	Het moet mogelijk zijn om objecten op te vragen gebaseerd op een of meerdere attributen.	Risico, toets, toetsplan, toetsplanning, constatering, bevinding, afwijking	Test	
0-006	Alle wijzigingen van de database moeten gelogd worden.		Test	M
0-007	Rapportages moeten geëxporteerd kunnen worden naar .pdf.		Test	M
0-009-1	Alle documenten van OG moeten door RRCB in SharePoint worden		Test	M

Figuur 73 'System/Subsystem Specification RRCB'

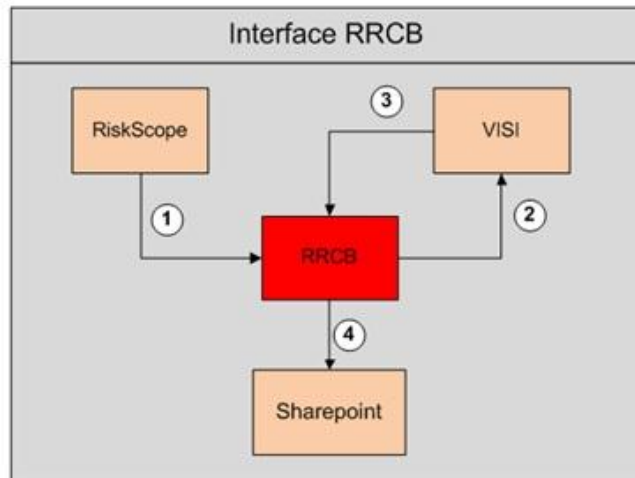
Interface Requirements Specification

De Interface Requirements Specification beschrijft de interface tussen RRCB, VISI, Riskscope en Sharepoint. Zie afbeelding 37. Het document is voornamelijk tekstueel opgebouwd en beschrijft in hoofdstuk 3 de eisen.

3.1 Interface identificatie en diagrammen

RRCB moet zorgen voor een geïntegreerd systeemlandschap, waarbij de gegevens gekoppeld zijn en zoveel mogelijk automatisch worden uitgewisseld tussen de verschillende systemen. Het aantal systemen waarmee de gebruikers te maken hebben is relatief groot. In figuur 1 wordt de relatie tussen RRCB en andere applicaties aangegeven.

I



Figuur 1 Interface RRCB

Figuur 74 'Interface RRCB'